

# Developing a Scoring Model to Evaluate Project Management Software Packages Based on ISO/IEC Software Evaluation Criterion

Hani M. Gharaibeh

PMPC Consulting Inc., Amman, Jordan.  
Email: [hani@pmpc-inc.com](mailto:hani@pmpc-inc.com)

Received August 11<sup>th</sup>, 2013; revised September 11<sup>th</sup>, 2013; accepted September 20<sup>th</sup>, 2013

Copyright © 2014 Hani M. Gharaibeh. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited. In accordance of the Creative Commons Attribution License all Copyrights © 2014 are reserved for SCIRP and the owner of the intellectual property Hani M. Gharaibeh. All Copyright © 2014 are guarded by law and by SCIRP as a guardian.

## ABSTRACT

Since the mid 1980s, an increasing number of project management software packages (PMSP) have appeared in the market to support project management organizations. Currently there are more than 500 software tools available. They vary widely both in their prices and capabilities. Despite the wide spread of PMSP, project-based organizations are left unguided as to how they should select the most appropriate software tool for their intended business use. The aim of this paper was to develop an evaluation/scoring module suitable for comparing different project management software packages to help select the most appropriate package. The evaluation model is based on the ISO/IEC evaluation criterion that takes into consideration all the necessary features and characteristics required in project management software. The evaluation model will provide software users and project organizations with a useful tool to compare various packages available in the market and select the best that suits their business needs.

## KEYWORDS

Software Evaluation; Software Characteristics; Attributes; Scoring Model; Cost Effectiveness Ratio

## 1. Introduction

Project management software is a set of software functions and tools specially targeted to support the management of projects. As such, the software attempts to provide planning, tracking, analysis, and reporting tools to software users. Today, project management software is viewed as a key element in the software kit; the differences from the past are the features and the ability to link to other softwares and employees through the network [1].

Project management software was initially developed in the 1970s to run on large computers. In the early 1980s, with the emergence of Personal Computers (PC), several packages for project management on a micro-computer emerged. The earlier project management software tools focused on scheduling and producing output, while analysis tools were minimal [1].

Software production companies are always under pres-

sure to deliver products that achieve greater functionality, higher levels of quality and greater ease of use. The software product must also be developed on time, within budget and with the user level of expectations being met. Needless to say, software developers rarely meet this level of expectations. One potential method for meeting the rising demands for software is to provide better tools to software engineers [1].

Project management software grew up in the 1970s on mainframe computers dedicated to large aerospace and construction projects. In the mid 1980s, software vendors began automating the techniques of PERT/CPM. The first of these packages cost approximately one million US dollars and ran on large computers containing numerous projects [2]. The advent of the desktop computers in the 1980s, and powerful inexpensive PC-based project management software encouraged the use of project management software tools in system development projects

[3].

The first project management software packages were based on PERT or CPM. They helped produce feasible project plans. Project management packages were then extended to record the resources needed to carry out the tasks, and hence indirectly the costs involved. Their use became standard practice in industries such as the construction industry in the 1950s.

Nowadays, there are thousands of project management software packages available for purchase, so that choosing the right package might not be an easy task. A fundamental reason for the extensive use of software tools is that they reduce a task to a routine procedure saving labor and eliminating a source of human error. Eliminating the source of error is the most effective way to improve productivity, as greater productivity improvements come from eliminating mistakes than that from performing tasks more efficiently [4].

Computer programs for project planning and control have been available for a long time, but the complex designs made general use difficult. The first generation of project management programs was complex, expensive, restrictive and difficult to use. However, new hardware and software technologies have opened new possibilities. The increasing popularity, acceptance and use of project management software are evidenced by a number of commercial programs that have been introduced in recent years. Many computer trade publications such as PC week, PC' World, Computer Shopper, and Software Magazine frequently carry advertisements, buyer's guides, and reviews on project management software [5] Brown, 1994.

Project management software is designed for use not only by the project manager, but also by any person involved in managing a group of resources in order to carry out a set of tasks. In large projects, the overall project leader cannot be aware of all the concurrent activities in a project. A decentralized organization of project management is often more helpful in coordinating the various activities in a large project. The project leader sets the objectives and time frames for accomplishing the project and then delegates the actual functions to different groups.

The major benefits of project management software are their ability to update reports, graphics and schedules when changes are required to be made. Drawing a large Gantt chart by hand could take several days, and one minor or major change involves redrawing the entire chart. Project management software gives the project manager more time to spend on pressing issues than on redrawing a Gantt or PERT chart.

A research survey conducted by Brown (1994) on the use of PMSP within Australian development organization showed that 77% of the respondents use some form of project management software to assist in performing

the activities of project management. PMSP was perceived by the respondents to be beneficial to the organization in terms of increasing productivity, software quality, techniques and simplifying activities.

The respondents of the survey were asked to name the actual software package that performs or assists in performing project management activities. The activities ranged from analyzing project risks, to defining WBS, developing schedules, and estimating project costs and size, assigning resources to reporting and analyzing on project activities. The results are described in **Table 1**. Among the 70 record responses, MS-Project is the most used PMSP, followed by Project Manager Workbench and CA super project [1].

As Gido and Clements (1999) note, there are numerous benefits of using project management software. Some of these are as follows [6].

**1. Accuracy:** A major benefit of using project management software is that accuracy is greatly improved. For large projects, manually drawing network diagrams, calculating start and finish times and monitoring resource usage are very difficult. Project management software packages have accurate algorithms for calculating project information and contain numerous built-in routines that check for user errors.

**2. Affordability:** Excellent PC-based project management software can be purchased for between \$400 and \$1200. This price might be high for an individual, but for most business such software is well worth the money.

**3. Ease of use:** Over the past few years, project management software packages have become extremely easy to use. They can often be mastered with only a minimal amount of training. This fact and the fact that most packages are affordable have led to a significant increase in the number of users of project management software.

**4. Ability to handle complexity:** It is obvious that software can handle certain aspects of projects well (especially numeric aspects). For projects that have just a few activities and span a short period of time, a manual

**Table 1. Degree of use of project management software.**

Package	No of customers	%
MS Project	30	49%
Time Line	5	8%
Project Manager-Workstation	12	20%
CA Super Project	7	11%
Primavera P3	2	3%
On Target	2	3%
Others	3	5%

approach may be feasible. However, for projects that have thousands of activities and thousands of resources and span a few years, project management software provides indispensable assistance with the level of complexity.

**5. Maintainability and modifiability:** With manual systems, it is often difficult to maintain and modify project information. For example, if a project is being managed without computer support, the network diagrams must be manually redrawn and costs must be recalculated every time when there is a change. With project management software, any change in data will automatically be reflected in all project documents, such as the diagrams, the cost tables, and resource allocation charts.

**6. Record keeping:** A major benefit of project management software is its ability to keep excellent records. For instance, the data can be kept on individual team members' schedules, as well as each task, cost and resources used. These data can be used to produce high-quality reports and can be helpful in planning future projects.

**7. Speed:** Once input data have been collected and entered, almost every imaginable calculation can be done very rapidly by the software. Manually creating or revising plans, schedules and budgets can take hours, days, or weeks. However, revisions can typically be performed in minutes or seconds with today's systems. The savings in time are usually enough to pay for the software itself.

**8. What-if analysis:** One further feature of project management software that is of great benefit is the ability to perform "what-if" analysis. "What-if" analysis allows the user to set the effects of various scenarios on a project. These different scenarios can be run on the software, and their effects can be evaluated. This allows the project manager to prepare and plan for certain contingencies and to assess consequences. Conducting "what-if" analysis without the software is not easy and sometimes impossible.

## 2. ISO/IEC14598 Software Evaluation Criterion

Today the use of information technology is growing; the number of critical computer systems is also growing. The quality of software in these systems is important because software faults may lead to serious consequences. Thus, it is important before starting the evaluation process to define, what is meant by the evaluation of project management software. It is the process of quantifying the different tools and features provided by a software package in a way that will help to compare it with different software packages.

Software products are extremely varied. They are written and produced to fulfill very diverse requirements.

Their context for use can also be very varied, such as in the case of application software in a management information system. The primary purpose of software product evaluation is to provide quantitative results concerning software product quality that are comprehensive and acceptable as well as depended on by any interested party.

The evaluation process is described as a step-wise procedure that allows expression of evaluation requirements in terms of quality characteristics, as defined in ISO/IEC 1996. The evaluation takes into account various documents that can be considered as part of the software production such as: design documentation, test or validation reports, and source code or user documentation. It is recommended by the ISO standards that the evaluator use a library of evaluation modules that define evaluation methods. These evaluation modules could be standardized, although no provision for that is proposed in this standard. The evaluation leads to the production by the evaluator, of an evaluation report [7].

There are different types of evaluation. The evaluation process can be classified by the party concerned with the evaluation, and the evaluation could be classified based on the stage at which the software is being evaluated [7].

The International Standards ISO/IEC 14598 series, classifies the evaluation process into the following three categories:

### 1) Process for Developers (ISO/IEC 14598-3)

This volume of the ISO standard should be used by organizations that are planning to develop a new product or to enhance an existing product using members of its own technical staff. It focuses on the use of those indicators that can predict end product quality by measuring intermediate products developed during the life cycle.

### 2) Process for Acquirers (ISO REC 14598-4)

This volume of the ISO standards should be used by organizations that are planning to acquire or reuse an existing or pre-developed software product. It can be applied for the purpose of deciding on the acceptance of the product.

### 3) Process for Evaluators (ISO/IEC 14598-5)

This volume of the ISO standards should be used by evaluators carrying out an independent assessment of a software product. This evaluation could be performed at the request of either a developer or acquirer or some other party. This part is intended for those who perform independent evaluation. Often they work for a third party organization.

The last type of evaluation is the focus of this investigation of evaluating project management software. The next section discusses in details this type of evaluation.

The purpose of this particular investigation is to provide a module to evaluating project management software. It is hoped that it will assist project management

organizations and general contractors in the process of selecting a project management tool that will meet their requirements and achieve the best performance in project planning and control.

### 3. Implementing the ISO/IEC 14598

The ISO/IEE 14598 provides a general method for software product evaluation that can be applicable to a variety of software products in different areas. This paper is particularly concerned with evaluating project management software packages, with which the ISO standards might not be directly concerned with. However, the evaluation process described in ISO/IEC 14598-1 was found to provide a well-structured and organized method that can be used as a guide through the evaluation process. Based on this view, the ISO/IEC 14598 will be implemented in the evaluation of project management software. However it will only be used as a general framework. A different evaluation module will be developed based on a decision analysis technique.

The start of the evaluation process includes the establishment of the evaluation requirements. This means that there is a need to specify the requirement of this evaluation as follows:

#### 1) Evaluation requirements:

The evaluation process intends to meet these requirements as follows:

1. The evaluation process shall lead to produce an evaluation module that is appropriate to be used to evaluate or assess project management software, by providing a quantitative measure for the characteristics of the software package.

2. The evaluation process shall involve the consultation of the software user as being considered a significant part of the evaluation process. The software user shall be consulted during the establishment of the evaluation criteria.

3. The evaluation process shall involve collaboration with the software vendors of the software packages that will be tested using the module. The evaluation shall include collecting data from software vendors that are relevant to the investigation.

4. The evaluation module shall cover all the aspects related to project management software, to make the evaluation efficient.

5. The evaluation module shall include numerical assessment of the various functions and tools provided in the software. This assessment shall be neither biased towards a particular software package nor affected by the feelings or emotions of the evaluator.

6. The evaluation module shall fulfill the characteristics of the evaluation previously mentioned. This is to

say that, the evaluation results for the same software package done by the same or by a different evaluator shall be consistent.

7. The data gathered for the purpose of the evaluation shall include a variety of users. In other words, all kinds of users that might be using project management software must be included in the evaluation to help obtain accurate results.

8. The evaluation shall include general information on the software use and applications by project management organizations, and the experience and knowledge of these organizations in engineering project management, as well as gathering general information on the software packages that will be used to test the evaluation module.

9. At the end of the evaluation process, the evaluator shall produce an evaluation report that summarizes all the findings of this investigation, and the final results of the analysis of the evaluation module.

#### 2) Specification of the Evaluation

Having established the evaluation requirements, the next step in the evaluation process is to determine the specification of the evaluation. This includes the following activities:

- Analyze the product description

The product that is examined in this investigation is basically project management software used to perform project management activities for the engineering industry. It is software used to plan, manage, organize and monitor engineering projects. This includes projects for the construction, manufacturing, chemical industries, etc.

- Specify the measurements to be performed

This stage involves determining the measurement to be performed in order to satisfy the evaluation requirements. The following characteristics will be measured in the software.

A. Operational Characteristics: It is the part of the evaluation that is related to the software capabilities and performance in terms of project management functions, tools and techniques. Under this category, the following sub characteristics will have to be measured:

- Time management: the capability of the software to organize time-schedules, perform scheduling calculations and identify activities shall be measured.
- Resource management: the ability of the software to identify resources, assign resources to activities and perform resource analysis shall be measured.
- Cost management: the ability of the software to calculate costs, perform cost analysis and cash flow graphics shall also be measured.
- Reporting: the ability of the software package to create and customize variety of reports on different types of data (schedule, resource and cost) will also be measured.

- Monitoring and control the ability of the software to manage projects, organize project data and communicate different information with project stakeholders will be measured.

B. Information Systems Characteristics: It is the part of the evaluation which is applicable to any type of software package, not just project management software, and it includes all the general requirements and characteristics that should be available in computer software. The following sub characteristics, in regard to any computer software package will have to be measured:

- User Friendliness
- Documentation
- Compatibility
- Technical Support
- Security

C. Life Cycle Cost Characteristics: It is the part associated with the different types of cost related to the software package through its life cycle, this helps to determine the total cost of the selected software package. The following cost items associated with the software will be calculated:

- Price of the software license
- Cost of installation
- Cost of hardware equipment
- Cost of maintenance
- Cost of upgrading
- Cost of learning

A detailed description of the major components of each of these sets of characteristic will be discussed later in the software evaluation criteria. The next step after the measurement has been specified is to verify the specifications of the evaluation process.

- Verify the specifications

Verifying the specifications produced with regard to the evaluation requirements means that the above measurements will be sufficient to perform the evaluation according to the requirements mentioned earlier. Based on these measurements, the evaluation process will achieve the objectives of the evaluation.

So far, the first two stages in the evaluation process have been discussed: establishing the requirements of the evaluation and specifications of the evaluation. The third stage will be the design of the evaluation, which will describe the procedure used to evaluate project management software. The design of the evaluation consists mainly of two parts:

- Establishment of a software evaluation criteria; and
- Building the structure of the evaluation module

### 3) Design of the Evaluation

In order for the evaluation process to be effective in evaluating the selected software package, a specific criterion has to be established which will form the basis for

the evaluation process. For the evaluation of project management software, all the different issues that could affect the quality or efficiency or even the degree of use of project management software have to be examined. Understanding what the software user expects the software to be is quite important to be taken into consideration when evaluating the software. Having said that, the software user needs to be consulted and given the chance to create his/her own evaluation criterion. To achieve this aim, a survey form was established and distributed to 115 organizations around Australia. The software user was given the chance to comment on the items included in the evaluation criteria, adding or deleting items. Additionally, the software users were asked to rank a degree of importance to each item in the evaluation criteria.

This evaluation criterion consists of three main categories. Each category was divided into specific sets of items called characteristics, each characteristic set was divided into sub characteristics and each set contained the least measurable unit in the criteria, called: Attributes. (ISO/IEC 14589).

This criterion was used to develop a questionnaire that would aid in the evaluation process of the software package. This questionnaire was sent to several users, to obtain information on the first characteristics which is the operational characteristic. A selected evaluation module would be used in addition to the questionnaire; to measure the performance of each software package against the different attributes listed in these criteria. Moreover, another questionnaire should be sent to the vendors of the software packages under investigation to obtain information necessary on both the general system characteristics and the cost life-cycle characteristics. The data obtained from both questionnaires will be analyzed with the use of a selected evaluation module.

This evaluation criteria was developed by referring to the Project Management Institute's software survey forms, and based on a criteria identified by Shtub and Bard in 1994. The evaluation criteria include the following categories:

**4) Operational Characteristics:** It includes the following sub characteristics:

- Time management characteristics: The features and tools associated with basic scheduling capabilities in terms of activities. It includes the following attributes:
  - Setting up the project: defining the project initial data;
  - Defining activities and precedence relationships: entering activities and connecting it with the logical relationships;
  - Identifying calendars: creating new calendars, modifying existing calendars, assigning calendars on activities.

- Network calculations: how the software calculates float, early and late dates.
  - Network graphics: the ability of the package to produce charts like PERT, Gantt, etc.
  - Resource Management: It is associated with the tools and features of resource planning. It includes the following attributes:
    - Resource Identification: how the software sets up resources, what sort of data is required;
    - Resource allocation: how resources are allocated on activities, what sort of data required in assigning resources;
    - Resource leveling: how the software performs the resource leveling, does it provide the ability to control the leveling options;
    - Type of resource graphics available: histograms, resource tables, etc.; and
    - Ability to handle resource over allocation: how the software deals with the problems of resource over allocation.
  - Cost Management: It is associated with the calculations and presentation of cost data.
    - Assigning cost data on activities: how to identify cost data in the software;
    - Support of cost breakdown structure: how the software structures the cost system;
    - Ability to perform cost calculations and Earned Value Analysis;
    - Support of uncertainty (risk analysis, what-if analysis, etc.); and
    - Cost graphics available: cash flows, cost profiles, etc.
  - Reporting: examining the reporting tools, and the ability to present different kinds of project reports. It includes the following attributes:
    - Type of standard reports available: schedule, resources, and cost;
    - Report generator and customization: the ability to create new reports and modify existing reports according to the user needs;
    - Report presentation: the ability to view reports on screen prior to printing; and control printing options;
    - Integration with Word processor: the ability to send and receive text files.
    - Controlling and Monitoring: The last part in the operational criterion is concerned with the features and characteristics that would help in controlling and monitoring the project plan, and tools that help in presenting and entering project data easily. This includes the following attributes:
      - Storing project baselines.
      - Outlining and data banding for activities and re-
    - sources data;
    - Master/subproject capabilities and multiple project management
    - Support of WBS (Work Breakdown Structure)
    - Web Publishing wizards, Internet tools
- 5) General Systems Characteristics**
- This includes the following sub characteristics:
- User Friendliness: the features and tools that will make the software flexible, easy to use, and deal with. This includes the following attributes:
    - Time to learn: the time it takes the ordinary user to learn the basic functions of the software package, if he or she is familiar with other project management principles;
    - Help wizard: the software should include help tools, on-line help;
    - Menu structure: the menu organization should be clear and the commands should be easy to understand and access;
    - Screen Messages: the screen messages and dialog boxes must be clear.
    - Multi-level of undo-redo to help with those unexpected and unplanned mistakes.
  - Documentation: is needed to provide written instructions or computer disks that explain and help understand and use the software package. This includes the following attributes:
    - User manuals: manuals related to the process of understanding, using and utilizing the software package in terms of project management functions.
    - Technical manuals: manuals that explain the installation and maintenance instructions of the software package.
    - Auxiliary Materials: educational materials that give general idea on the software application, features, performance and capabilities. This includes: Demo CDs, brochures, booklets, etc.
  - Compatibility: the ability of the software to communicate and interact with other information systems in regard to data exchange and safe operations. These characteristics include the following attributes:
    - Microsoft Office applications: the ability to communicate and exchange data with different office applications such as: Word, Excel, Access and Power Point.
    - Other project management software: the ability to exchange files and data with other project management software; and
    - Other information systems: the ability to work friendly with other systems like: database management systems, programming languages.
  - Technical Support: the service and consultancy pro-

vided by the software vendor after the software being purchased by the customer. This includes the following attributes:

- Implementation: the process of installing and operating the software package at the customer's workplace;
  - Maintenance: technical advice and support by the software vendor in terms of any operational problems or problems related to the performance of the software; and
  - Training: whether or not the software vendor provides training courses, lectures, seminars, etc. on the use and application of the software package.
- Security: It is the ability to provide secure access system in regard to the following attributes:
- Security on baseline plans: password required to access the baseline
  - Security on accessing project files: password and user name required to access the software files.

#### 6) Life—Cycle Cost Criterion

This criterion includes calculating the following cost items:

- Price of the software license: the cost of the software package per one license including all the supporting materials.
- Cost of hardware: the cost of hardware equipment that will provide the minimum hardware requirements to operate the software package. (HDD, CD-ROM, etc.);
- Cost of installation: the service fee charged by the software vendor to install one package of software at the customer's workplace.
- Cost of maintenance: the cost of technical services and advice on the software, regarding any problems in the performance of the software, provided no hardware parts are to be installed.
- Cost of upgrading the software package from old to new version; and
- Cost of learning: how much does it cost the ordinary user to learn the basic functions of the software package, assuming he or she is familiar with project management principles.

#### 7) Execution of the Evaluation—The Scoring Module

Having established the evaluation criteria, an evaluation module is then selected. The evaluation module is a tool that will provide quantitative assessment of the different functions of the software package, covering all the aspects and attributes identified in the evaluation criteria.

There are several types of techniques/modules that have been used in the process of evaluating computer software. One of the most popular techniques is the Capability Maturity Module or CCM. This is a fairly old

technique developed in the US. However the ISO/IEC 14598 provides a useful technique for software evaluation as discussed earlier. Another significant evaluation method used in this area is the Decision Analysis technique. This technique is popular in the area of business management and marketing. It was developed to help managers and decision-makers to choose among different alternatives and thus selecting the right decision.

Several decision analysis techniques have been developed to assess and evaluate various alternatives; each technique has its own methodology, and the decision to select a particular technique, very much depends on the type of data collected. Among these techniques are Simple Multi Attribute Rating Technique (SMART), Analytical Hierarchy Process (AHP), decision trees, decision tables and sensitivity analysis.

When it comes to project management software evaluation, it is not an easy task to find the appropriate technique that would be suitable to assess project management software. A well-known decision analysis technique that has been used to evaluate and study project management software is the Scoring module.

A scoring module is a simple decision analysis table that uses scores assigned to each attribute in each software package based on the performance of this particular package against certain criteria items. The scoring module provides a quantitative assessment of the software package by providing scores and weights for all the attributes in the software. The final score measures the overall performance of the software package in respect to the evaluation criteria.

The advantages of using the scoring module are:

- a) The provision of a quantitative measurement of the overall performance of the software package, which helps to distinguish among several packages;
- b) The coverage of a wide range of various attributes to be evaluated in the software package;
- c) The simplicity and the fact that it does not require the use of any special formulas or complex mathematical calculations;
- d) The type of data collected to perform the evaluation is simple and easy to obtain;
- e) It takes into consideration the different degrees of importance for the different attributes in the evaluation process, and hence, gives more accurate results; and
- f) It is flexible in the way the tables are formed and gives complete freedom in choosing the attributes to be included in the evaluation.

The scoring module is created based on the software evaluation criteria discussed in the previous section. For each of the characteristics and attributes described previously, the following will be calculated:

- Score: a numeric value assigned to the software pack-

age, to measure the performance of this package against a particular attribute. This score will be assigned based on the scale as shown in **Table 1**.

- Weight: a percentage value (out of 100%) used to express the degree of importance of each particular attribute in respect to the specified sub characteristic set examined in the evaluation criteria. This value will be assigned to each attribute on the basis that the sum of the weight for each sub characteristic set should equal 100, that is the total weight of each table should equal 100%;
- Relative weight (X): the weight associated with each sub characteristic set with respect to the selected evaluation criterion (the total weight in each characteristic set.
- Relative weighted score (Y): the sum of the weighted scores of each characteristic set.
- Weighted score: the score of each package against each particular attribute, multiplied by the degree of importance of each attribute. This is calculated according to the following formula.

$$\text{Weighted Score} = \text{Score} * \text{Weight} \quad (3.1)$$

The sum of all the weighted scores for each characteristic set for a selected package will give the total score of this package for this particular set of characteristics. This should be then multiplied by the relative weight of each characteristic set. Adding all this up will give rise to the weighted score for each package against the three main characteristics examined. This is then multiplied by the criteria weight, and the sum will give the relative weighted score for each sub characteristic set.

$$\begin{aligned} &\text{Relative Weighted Score Y} \\ &= \sum (\text{weighted scores/sub characteristic set}) \end{aligned} \quad (3.2)$$

The following equation will give the total weighted score of the software package in respect to the operational characteristics:

$$\begin{aligned} A = &X1 \cdot Y1 + X2 \cdot Y2 + X3 \cdot Y3 \\ &+ X4 \cdot Y4 + X5 \cdot Y5 = \sum_{i=1}^5 X_i \cdot Y_i \end{aligned} \quad (3.3)$$

where:

X1: Relative weight of the time management sub-characteristics

Y1: Relative weighted score of the time management sub characteristics

X2: Relative weight of the resource management sub characteristics

Y2: Relative weighted score of the resource management sub characteristics

X3: Relative weight of the cost management sub characteristics

Y3: Relative weighted score of the cost management

sub characteristics

X4: Relative weight of the reporting sub characteristics

Y4: Relative weighted score of the reporting sub-characteristics

X5: Relative weight of the monitoring and control sub characteristics

Y5: Relative weighted score of the monitoring and control sub characteristics.

The following equation will describe the total weight of the software package in respect to the general information characteristics:

$$\begin{aligned} B = &X6 \cdot Y6 + X7 \cdot Y7 + X8 \cdot Y8 \\ &+ X9 \cdot Y9 + X10 \cdot Y10 = \sum_{i=6}^{10} X_i \cdot Y_i \end{aligned} \quad (3.4)$$

where:

X6: Relative weight of the user documentation sub characteristics

Y6: Relative weighted score of the user documentation sub characteristics

X7: Relative weight of the compatibility sub characteristics

Y7: Relative weighted score of the compatibility sub characteristics

X8: Relative weight of the security sub characteristics

Y8: Relative weighted score of the security sub characteristics

X9: Relative weight of the technical support sub characteristics

Y9: Relative weighted score of the technical support sub characteristics

X10: Relative weight of the user friendliness sub characteristics

Y10: Relative weighted score of the user friendliness sub characteristics.

The final equation describes the total weighted score of the software package in respect to the examined criteria.

$$C = A * m + B * n \quad (3.5)$$

where:

C: The total weighted score of the software package.

m: The relative weight of the operational characteristics

n: The relative weight of the general information characteristics.

However, this method will only be applied to both the operational and general system characteristic, the criterion related to software costs will not be included since a cost effectiveness measure will be used as part of the selection process.

The scoring model will use the following terminology to describe the score of the software package against each

single attribute:

- Poor: If the attribute (see **Table 2**) does not exist in the software package (zero value, minimum score);
- Bad: If the attribute exists, but is presented badly in the software (There are weak points, major difficulties in the use of the attribute);
- Good: if the attribute is achieved within minimum requirements (average performance);
- Excellent: If the attribute is performed very well in the software, easily, in a flexible way and with extra capabilities.

The cost criteria will be dealt with separately by calculating all the items of cost associated with the software package, as indicated in the evaluation criterion, and at the end we will come up with the total cost of each software package. To obtain the data necessary for the cost calculations, as we mentioned in the criteria we have to question software vendors about certain items related to cost. The cost in this case will include the items listed in **Table 3**.

After obtaining the total cost of each software package separately and having calculated the total score for each package, we can calculate the cost-effectiveness ratio as follows:

$$\text{Effectiveness Ratio (points / \$)} = \frac{\text{Total weighted score (points)}}{\text{Total cost (\$)}} \quad [8, \text{ p. } 567]$$

The effectiveness ratio is basically a measure of the overall performance of the software package against all the examined criteria items. It indicates the extent to which the software package is efficient and capable of meeting the evaluation criteria. The higher the effectiveness ratio, the better the software will be in terms of the overall performance.

Base on the characteristics, sub characteristics and attributes examined, the scoring module is established. **Table 4** represents the structure of the scoring module discussed above.

#### 4. Analysis of the Software User Survey Form

To help create the evaluation module, a survey was needed, within the Australian construction industry, to collect the information necessary to build up the scoring module. A random sample of 115 organizations with different specialties was selected. The number of respondents was 45 representing a response rate of 39%. The sample includes architects, contractors, and engineering designers, as well as project management organizations. The organizations were selected from different cities in Australia, to try to ensure that the sample was a repre-

**Table 2. The Scale used in the scoring model.**

Score	Performance	Condition
0		
1	Poor	If the attributes does not exist at all in the software package
2	Bad	IF the attribute exist, but presented badly (there are minor problems, weak points in the use of the attribute)
3	Good	IF the attribute is achieved within minimum requirements or average performance
4	Excellent	If the attribute is expressed very well with powerful tools provided in this attribute

**Table 3. Cost breakdown of software items.**

Cost Item	Unit
Purchase cost of the software	
Installation cost	
Maintenance cost	
Upgrading cost	
Hardware equipment cost	
Learning cost	
Total cost of the software	

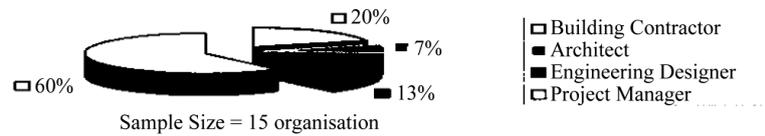
sentative the of Australian construction industry. The cities chosen were Sydney, Melbourne, Brisbane, Adelaide and Perth. The companies chosen also varied in their size, ranging from small and medium to large. The value of projects undertaken by those companies ranged from one Million AUS dollar to more than 50 Million AUS \$/yr.

Within this sample, it was found that the majority (60%) was project management organizations (**Figure 1**). This value indicates the importance of the use of PMSP for this type of organizations. Since project management is the core of their business, these organizations depend on software to perform so many tasks. With only 13% architectural organizations in the sample, it appears that PMSP is of less concern to architectural design firms. The second type of organizations interested in PMSP was the contractors, with 20% of the sample being as building contractors. The engineering designers tended to use PMSP less than the project managers and the contractors; with 7% of the sample size were designers. In general, it can be concluded that the more the organization is involved with project management activities, the more they are likely to use PMSP.

As mentioned earlier, the sample included organizations with different sizes in terms of number of employees. This makes the sample a representative one and it covers a wide range of organizations in terms of size. Among the responding organizations 20% employed more than 100 employees, those were considered large

**Table 4. The scoring model according to the ISO/IEC software evaluation criteria.**

Software evaluation criteria	Weight	Package A		Package B	
Operational characteristics	0% - 100%	Score	Weighted	Score	Weighted
Software attributes		Score		Score	
Time management relative weight X1 =					
Setting up your project plan					
Defining activities and relationships					
Calendars					
Network calculations					
Network graphics					
Relative weighted score Y1					
Resource management relative weight X2					
Resource identification					
Resource allocation					
Resource allocation					
Resource levelling					
Ability to handle resource overall action					
Type of resource graphics available					
Relative weighted score Y2					
Cost management relative weight X3					
Assigning cost information					
Supporting cost breakdown structure					
Cost calculations					
Uncertainty analysis					
Cost graphics					
Relative weighted score Y3					
Reporting relative weight X4					
Type of standard reports available					
Report generation and customization					
Integration with word processor					
Report presentation					
Relative weighted score Y4					
Control and monitoring relative weight X5					
Saving project baseline plans					
Outlining and data banding					
Support of WBS					
Multi project management					
Web publishing and internet tools					
Relative weighted score Y5					



**Figure 1. Type of the respondent organization.**

organizations. The majority of the organizations were medium and small in size (80%), employing less than 100 people for each organization (see [Figure 2](#)).

When asked about the type of project organization, 60% of the users were building contractors. Only 7% were Architects. This shows that the use of project management software becomes more apparent as the level of involvement in the project increases ([Figure 3](#)).

When the users were asked whether the availability of an evaluation module would help them in the evaluation of PMSP, 73% agreed that using an evaluation module would be a good method to evaluate PMSP. This result indicates how important the outcome of this study will be for software users and organizations. Only 7% disagreed with the need to use an evaluation module in the process of project management software evaluation (see [Figure 4](#)).

The value of the projects undertaken by different organizations ranges from less than one million AUS \$/year to more than 50 million AUS \$/year. This also makes the sample a representative one for a wide range of organizations in terms of financial capability. The majority of the organizations undertaking projects were valued between one to 50 million AUS \$/year. Out of the 45 respondents to this questionnaire only 15 answered this question from which only one organization is undertaking projects with a value less than one million AUS \$. Fifteen organization out of 45 work with more than 50 million AUS \$/year and all indicated that they use project management and that more than 10 planners work with them. This shows the significance of the use of PMSP for huge projects and within financially large organizations (see [Figure 5](#)).

In relation to the type of PMSP used among respondents, only 24 organizations answered this question from the 45 respondents. Among which MS-Project achieved superiority over other types of PMSP. With 12 organizations using MS-Project it remains the most popular and commonly used PMSP. Primavera P3 was only used by 6 organizations (see [Figure 6](#)).

The number of software licenses used in each organization was found to be related to the size of the organization and the value of the projects undertaken. Fifteen out of 45 organizations who answered this question use more than ten software licenses, those organizations were medium to large in size, they undertook projects valued above one million AUS \$/year and sometimes above 50

million AUS \$/year. Only five organizations used one software licence, most of those organizations were small in size, they undertook projects valued at less than 10 million AUS \$/year. These highlights the increase in PMSP use as the size of the organization increases, and as the value of the projects gets higher. This need for a PMSP can be seen much more important in large projects than small sized projects ([Figure 7](#)).

In terms of experience with the use and application of PMSP, ten organizations indicated that they have more than 10 years experience with the use of PMSP. All of the ten were found to be project management organizations. It can be concluded that project managers are the most interested people in this kind of business software ([Figure 8](#)).

A surprising result of this survey was the number of planners in each organization. Six organizations indicated that they do not have any planners in their organization. ([Figure 9](#)), and yet, five of them mentioned project/construction management as one of their activities. This may lead to the conclusion that performing project management activities and services are not necessarily only done by planners.

The survey revealed that the main purpose of using PMSP was only to produce project time schedules ([Table 5](#)). It was found that fourteen organizations use the software for this purpose. The second most important purpose for users was monitoring and updating project plans. Meanwhile, only two organizations indicated that they use the advanced features provided by the software such as: web-publishing wizards, What-if analysis, earned value analysis. The user seems to be more concerned with producing simple charts and timetables rather than the complex reports or the sophisticated tools that the software provides.

It was also found that software users tend to be affected by other user opinions on the selection of the software. [Table 6](#) shows that 8 organizations selected their software packages because someone recommended this particular package to them. Some users prefer to test several packages on real project cases before selecting the appropriate one.

However, when the users were asked about the method to be used in the future, for selecting a new PMSP, the majority preferred to experience the use of a tool/package on a real project ([Table 7](#)). Also 10 organizations indicated that they would study the experience of other

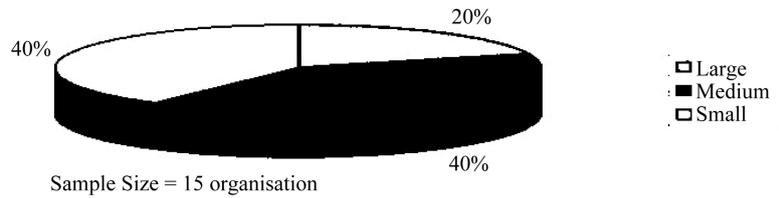


Figure 2. Size of the respondent organization.

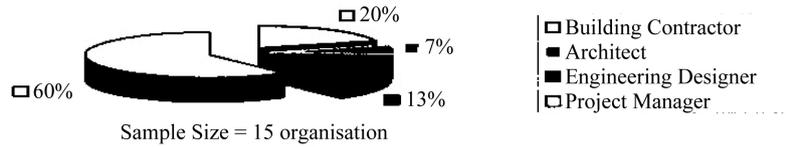


Figure 3. Type of the respondent organization.

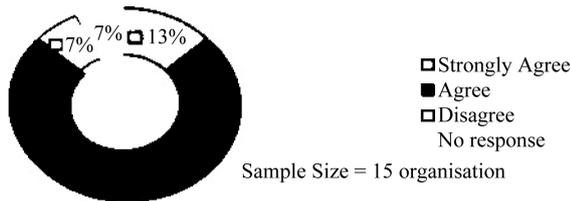


Figure 4. Respondents opinion on the use of evaluation model.

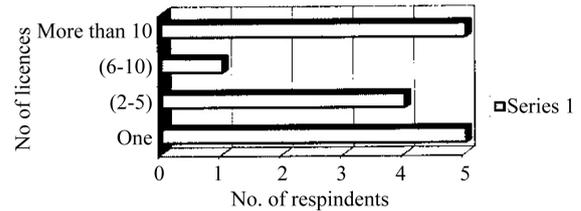


Figure 7. Number of software license used by the respondents.

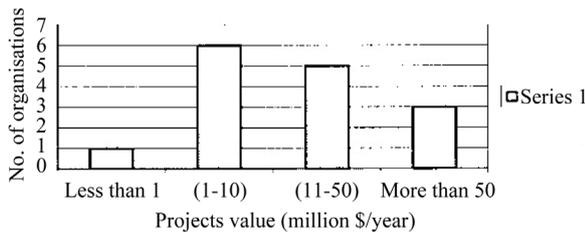


Figure 5. Total value of projects undertaken by the organization.

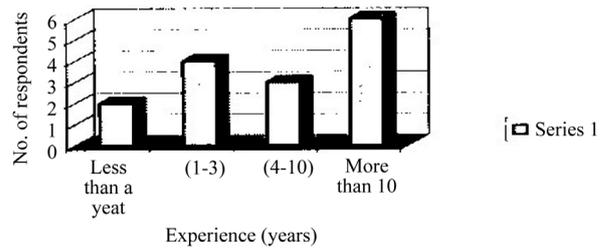


Figure 8. Respondent's experience with the software.

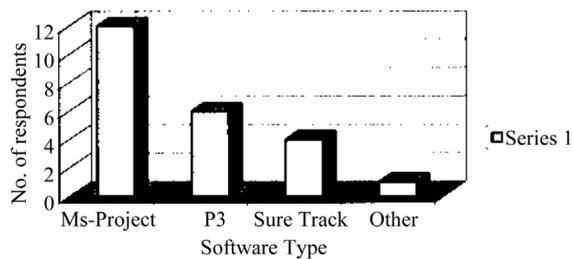


Figure 6. Type of PMSP used among respondents.

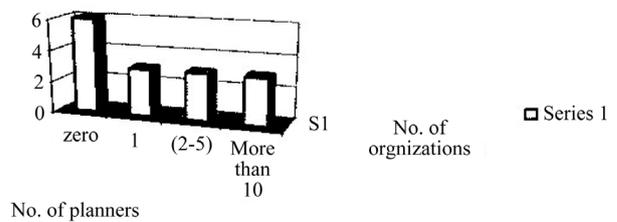


Figure 9. Number of planners in the respondent organization.

companies and users in this regard before making a selection.

### 5. Analysis of the Scoring Module

Going back to the scoring module, the analysis of the module starts by calculating the weights and scores of the different attributes. The first step in the analysis is to

determine the relative weights of both the operational and general system characteristics (m, n). For this particular section of the questionnaire, only 14 organizations provided their weight % assessment for each of the criteria items. Table 8 shows the relative weights assigned by each contacted organization. The figures indicate that the value assigned by the user is independent of the type or size of the organization or even the value of the projects

**Table 5. The purpose of using PMSP as indicated by the respondents.**

Purpose	No. of respondents
Produce Project time schedules	14
Manage your resources and project cost data	6
Update the project plan and monitor progress	10
Produce regular reports on different project data	5
Use advanced features in the software	2
Others	2

**Table 6. Software selection by respondents.**

How the respondent choose the software	No. of respondents
Someone recommended this software for you	8
The software vendor has an outstanding reputation	1
The most commonly used software package	1
Client/ Industry Performance	3
You tested several packages on real projects	4

**Table 7. Selection method used by the respondent to select a new PMSP.**

Selection method to be sued in the future	No of respondents
Consult project management organization or professional consulting group	3
Experience in the use of the tool on real projects	10
Studying the experience of others	7
Application of a sophisticated software evaluation model	2
In house training on a number of projects/ users	1

undertaken. It was found that each user gives different figures, and therefore, no relationship can be made between the parameters in this table. Having said that, the average weight will be taken to represent the user selection for the values of (m, n) since the sample is both small and relatively mixed.

The second step involves calculating the relative weights of all the sub characteristics in the module. **Tables 9 and 10** show the relative weights assigned by each user to the different sub characteristics in the evaluation criteria. Again the average weight is taken to represent this small sample. The values obtained are referred to as X1, X2, and X3... X10 in the scoring module.

The best PMSP, according to this module will score the maximum value on all attributes, assuming the software satisfies all the attributes fully and achieves excellent performance. It must be remembered that the worst PMSP will not score zero for all the attributes, because zero means that the attribute does not exist at all. A package that represents the minimum acceptable level of

**Table 8. Weights assigned by users for the software evaluation criteria.**

Type of organization	Operational	General system	Size of Org	Value of projects
				Million \$/yr.
Project management				
Project management	50	50	small	1 - 10
Project management	53	47	11 - 5-	11 - 50
Project management	50	50	small	1 - 10
Project management	80	20	med	More than 50
Project management	67	33	Large	11 - 50
Project management	56	44	medium	More than 50
Project management	57	43	medium	1 - 10
Project management	57	43	small	More than 50
Project management	67	33	small	Less than 1
Building contractor	57	44	medium	11 - 50
Building contractor	73	43	medium	11 - 50
Engineering designer	57	27	Large	1 - 10
Engineering designer	60	43	Large	More than 50
Average relative weight man	60	40		

**Table 9. Operational characteristics weight by organization.**

Respondent No	Operational characteristics weight %				
	Time	Resources	Cost	Reports	Control
1	80	0	0	10	10
2	60	10	0	0	30
3	20	20	20	20	20
4	30	20	15	30	5
5	24	19	14	21	22
6	34	27	20	13	6
7	21	21	16	21	21
8	21	18	21	20	20
9	30	10	20	20	20
10	50	10	10	15	15
11	20	20	20	20	20
12	25	16	20	20	20
13	21	21	16	21	21
14	30	5	5	30	30
<b>Relative Weight X1-5</b>	<b>33</b>	<b>15</b>	<b>14</b>	<b>19</b>	<b>19</b>

**Table 10. General system characteristics weight by organization.**

Respondent No	Operational characteristics Weight %				
	User Friendliness	Documentation	Compatibility	Security	Technical Support
1	80	10	10	0	0
2	70	0	0	0	30
3	42	42	10	3	3
4	55	15	5	5	10
5	27	24	22	19	8
6	34	27	20	13	6
7	22	22	20	16	20
8	21	18	20	20	21
9	50	20	10	10	10
10	38	12	12	0	38
11	23	23	18	18	18
12	33	22	28	6	11
13	29	21	15	6	29
14	30	30	20	10	10
<b>Relative Weight X6-10</b>	<b>40</b>	<b>20</b>	<b>16</b>	<b>9</b>	<b>15</b>

quality will score one point on all attributes.

### 6. Conclusions

The study revealed that there is not one best project management software product in the marketplace. Technology has enabled new functionality that may or may not help the organization meet its business goals and manage projects better. More sophisticated tools do not guarantee success. It must be remembered that the product that offers the most features is usually the most difficult to implement, learn and support. As seen from this study, these attributes are: implementation, time to learn, and technical support which are also important for the software user.

The process of evaluating PMSP should involve more than just the evaluation of the software package itself. A successful PMSP evaluation will include a comprehensive look at the organization and the processes. The evaluation process that an organization goes through selecting a particular package should include the environment, in which the software will be used, as well as the organization needs and objectives for implementing a PMSP. Implementing a software package without taking these issues into consideration means that the wrong package could be selected. This could become a problem for the whole organization.

The analysis of the scoring module revealed that, in terms of operational characteristics of a software package, time management is the most important function provided by the software, with a relative weight of 33%.

Among the different attributes of time management, setting up the project plan and defining activities and relationships were the most important attributes. This outcome seems to be consistent with the results obtained earlier from the user survey form, at which we found the main purpose of using a PMSP for the user is to produce time schedules.

The evaluation criteria examined the software in three areas: the operational characteristics, the general system characteristics and the cost of the software package. Among these factors, the operational characteristics of the software, which represents the functions and tools in the software were the most important, with a relative weight of 43%. Nevertheless, the cost was also a significant part in the evaluation. However, the cost accounts only for 24% of the evaluation criteria.

The scoring module in its simple structure provides an easy way for evaluating a particular PMSP by examining a comprehensive list of software attributes. These attributes are indicated by the actual user as being important in the software selection process. The method of assigning scores for each attribute helps provide a quantitative measure for the different features of the software. Using the simple equations of the scoring module will enable the calculation of the total weighted score of the software, which is a measure of the overall performance of the package. Once this is achieved, the efficiency of the software package can then be calculated. This outcome will help when comparing one package with the other package. Furthermore, the cost effectiveness ratio will be the final and true indication of the ability of the software

package to satisfy all the criteria items. The package with a higher CER should then be selected.

### REFERENCES

- [1] P. L. Bennet and R. R. Kathryn, "Project Management for the 21st Century," 3rd Edition, Academic Press, Waltham, 2002.
- [2] J. Connell and L. Shafer, "The Professional User's Guide to Software," 1987.
- [3] J. S. Edwards and T. I. Moores, "A Conflict in the Management of Information," 1994.
- [4] W. S. Humphrey, "Managing the Software Process," Mass Adisser, Wesley, 1989.
- [5] S. Brown, "Project Management Software Tool Selection and Usage," 1994.
- [6] J. Gido and J. P. Clements, "Successful Project Management. South Western," 1999.
- [7] ISO/IEC DIS 14589-1, "Information Technology-Software Product Evaluation. Version 7.6," International Standard Organization, 1996.
- [8] A. Shtub and J. F. Brand, "Project Management, Engineering Technology and Implementation," Prentice Hall Inc., Upper Sadder River, 1994.