8-15-1997

MASET: Multi-Attribute Software Evaluation Tool

Robert T. Keim
Arizona State University, robert.keim@asu.edu

Albert Kagan
Arizona State University

Gerald Post
Western Kentucky University

MASET: Multi-Attribute Software Evaluation Tool

Robert T. Keim[1] **Albert Kagan**[2] **Gerald Post**[3]
1 School of Accountancy & Information Management, Arizona State University, Tempe, AZ 85287-3606,
Phone 602/965-4445, Fax: 602/965-8392, e-mail: Robert.Keim@asu.edu
2SABER, Arizona State University, Tempe, AZ 85287-8206, Phone 602/965-5334
3 Department of Finance & CIS, Western Kentucky University, Bowling Green, KY 42101

Abstract

Software developers and organizational users face many problems in identifying and evaluating features of software products. The products are complex, with numerous options, making design and development of applications time consuming and costly. With many users, it can be difficult to summarize user needs and priorities, let alone identify the functions and features that they need in the applications. Once functions and features are identified, a three stage least squares technique is used to identify and prioritize important features of software products. The Multi-Attribute Software Evaluation Tool (MASET) can be used to evaluate existing software (commercial or in-house) or proposed features of new software products. This tool combines a series of functions into a software product that aids in the development of an evaluation questionaire, accepts input from respondents, prepares the input for the statistacal analysis package and uses an artifical intelligence module to analyze the results.

Introduction

Whether developed in-house or purchased from commercial vendors, software packages are becoming increasingly complex. These packages have many components with hundreds of features and options. Adding to the complexity, there are several competing packages available for most tasks. Both commercial and in-house production software offer different features and options with competing strengths and weaknesses.

Software buyers need to sort through all product features to select an appropriate package that will address the needs of the organization. In general, these trends toward expanding the features and functions of the packages are beneficial to purchasers and users of software. When designing a product, the developer is faced with determining which features are most important and the effectiveness of each. A preliminary needs assessment must first identify what these functions are and how to describe them. Developers must also compare proposed custom software to existing commercial packages. Often the user needs can be satisfied with commercial software. It may be cheaper to buy a package instead of incurring the costs of building and maintaining a custom system.

The problem that needs to be addressed in this assessment process is how to rank the features and, further, how to evaluate the individual features in the proposed product. MASET (Multi-Attribute Software Evaluation Tool) accepts as input the derived functions and features from the preliminary needs assessment and creates a questionnaire for distribution to the target users. After accumulating the results of the survey, the three stage least squares model is applied and the output is evaluated for the selection team.

In typical evaluation situations, without an objective and interactive assessment approach, a number of issues arise because of the complexity in evaluating software. These include: (1) Users disagree with each other over the value of different features. (2) Software designers must identify potential new features and choose directions for software. Because of the lack of specificity of the potential functions it becomes difficult to select the appropriate mix of features. (3) Summary or aggregate evaluations of software can miss crucial detailsóimplying the need for a more detailed evaluation process. (Henderson, et. al. [1995], Kitchen, et. al.[1996])

A common problem in these situations is the necessity to identify the relative importance of the multiple attributes of various software packages across multiple users. This method systematically utilizes individual

evaluations to determine an overall importance or ranking of the various attributes. This approach captures the opinions and needs of all relevant users.

Multi-Attribute Methods

There is considerable agreement that user involvement in software design and selection are important factors in producing useful systems (Keil and Carmel [1995]). Traditional design methodologies use interviews to develop an understand-ing of user tasks. Design problems are especially acute when there are many users involved in a project. Traditional design methodologies require users to resolve differences by negotiations among departments culminating in the sign-off on various decisions by steering committees.

Multiattribute decision-making described in the management science literature (Dyer et al. [1992]) has characteristics in common with the evaluation of software features. As software becomes more complex and multifunctioned, the evaluation of the parts, as well as the whole, must be considered when designing or selecting a product. The multiple attribute approach will allow the simultaneous evaluation of the whole and the parts while prioritizing the importance of the parts and determining the effectiveness of the individual part. The choice of methodology depends on the goals, data, and characteristics of the problem. One additional characteristic separates software design decisions from common multiattribute models: the importance of considering opinions relative to software needs and requirements from many different users.

Models

In many respects, software design and evaluation is similar to traditional product design. Aside from the physical nature of the software product, perhaps the largest difference between software and consumer product design is the complexity of the product and the number of features involved. Several models have been developed in the marketing literature to identify consumer preferences. These models are relatively easy to administer and can be applied to analysis of existing or proposed products. Such an approach is relevant to the design of software in that the features needed by users will become the basis for the design process.

Two types of techniques are useful: ordinal regression and structural equation modeling. The difference in the two approaches arises from the data that is available. If users are familiar with multiple products, then ordinal regression techniques can be used to identify the strongest attributes. If users are familiar with only one product or proposal, then structural equation models can be used to determine the relative importance of each product attribute.

The essence of the models is to describe user choices in terms of multiattribute utility. The various techniques analyze the user response data to determine the weights or part-worth valuation placed on each attribute.

For several reasons, it is useful to divide the software attributes into categories. The categories make it easier for users to understand and evaluate the software by dealing with smaller amounts of information at one time. Additionally, by asking users to evaluate categories as well as detail, the responses can be examined for consistency and reliability. More statistical information about the usersí true preferences can be ascertained through detailed process of product attribute categories.

Summary

MASET accepts the characteristics of the product as input, submits the data to this analysis, and outputs an evaluation of the functions and features of the software under consideration. It will provide the evaluation team with a realistic and unbiased evaluation of the application or product.

References

Dyer, J.S., Fishburn, P.C., Steuer, R.E., Wallenius, J., and Zionts, S., 1992, Multiple Criteria Decision Making, Multiattribute Utility Theory: The Next Ten Years, Management Science, 38(5), May, 645-746.

Henderson, R.D., Smith, M.C., Podd, J. and Varela-Alverez, H., A Comparison of the Four Prominent User-Based Methods for Evaluating the Usability of Computer Software, Ergonomics, 38(10), October 1995, 2030-2044.

Keil, M. and Carmel, E., 1995, Customer-Devloper Links in Software Development, Communications of the ACM, 38 (5), 33-44.

Kitchenham, B, and Pfleeger, S.L., Software Quality: The elusive Target, IEEE Software, 13(1), January, 1996, 12-21.