USING THE ANALYTIC HIERARCHY PROCESS TO EVALUATE SOFTWARE DEVELOPMENT METHODS FOR INFORMATION SYSTEMS CURRICULUM

Mary J. Granger
Assistant Professor
George Washington University

ABSTRACT

"A Software Development Method (SDM) is a system of technical procedures and notational conventions for the organized construction of software" (Karam and Casselman 1993, p. 34). The two most popular SDMs are Information Engineering and Software Engineering.

Karam and Casselman provide a framework for evaluating SDMs within an organization. The purpose of this study is to determine which SDM, Information Engineering or Software Engineering, is appropriate for incorporation into the Information Systems curriculum. Faculty from several different universities completed Analytic Hierarchy Process-based surveys to evaluate the SDMs. It is anticipated that the Analytic Hierarchy Process will facilitate future curriculum decision making, especially decisions involving software development methodologies.

KEY WORDS:
software, development, methodology, Information Systems, curriculum, framework, education

INTRODUCTION

Software developers are constantly exploring new software development methods that are timely and cost effective, and also foster productivity and better quality systems. Information Systems faculty are also constantly searching for the ideal software development method for incorporation into the Information Systems curriculum.

"A Software Development Method (SDM) is a system of technical procedures and notational conventions for the organized construction of software." (Karam and Casselman 1993, p. 34) The introduction of Computer-Aided Software Engineering software has generated attention toward a variety of software development methodologies. Therefore the selection of an appropriate SDM can be a major decision for software developers within a business Information System. Business organizations are re-evaluating SDMs, therefore IS faculty teaching software development methodologies also need to perform a similar evaluation.

IS graduates with an understanding of the business organization and a knowledge of SDMs are highly recruited by business organizations. IS faculty are trying to determine the best curriculum for their students and their choice of a SDM requires serious consideration. However, "The results tend to indicate that in spite of two decades of experience in developing software, the question of the best methodology is still largely unsettled" (Keyes 1992).

Currently, the two most popular SDMs are Information Engineering and Software Engineering. Karam and Casselman (1993) provide a framework for evaluating SDMs within an organization. There are 14 technical properties, 5 usage properties and 2 managerial properties. The purpose of this study is determine which SDM, Information Engineering or Software Engineering, is appropriate for incorporation into the Information Systems curriculum. Faculty teaching software development and design at different universities completed Analytic Hierarchy Process (AHP)-based surveys to evaluate the SDMs. The results are reported in this paper.

INFORMATION SYSTEMS CURRICULUM

Information Systems have changed the way the organization conducts business. It is important for Information Systems graduates to know state of the art techniques, and therefore, courses must constantly be updated to meet this demand. Current IS students will be the system analysts of tomorrow, therefore, they need to understand at least one SDM.

The number of courses in an IS curriculum is constrained by accreditation and university policies. Although there is at least one project oriented course, most often at the senior level, the choice of SDM may influence other IS courses thereby impacting the whole IS curriculum. The major focus of this course is the SDM or SDMs which will be used to guide the analysis, design and possibly the implementation of a project. The student project(s) within this course is limited in size by the semester/quarter time, available computer resources and a learning curve for the SDM. Too often the availability of Computer-Aided Software Engineering (CASE) tools dictates the SDM introduced in the project oriented course. Using Karam and Casselman's framework for evaluating SDMs in conjunction with AHP facilitates the SDM decision.

SOFTWARE DEVELOPMENT METHODOLOGIES

Software development methodologies provide a logical representation of a software system. Unfortunately many SDMs are imprecise, incomplete, over-prescriptive, static, tool oriented and lack quality assurance procedures (Bloor 1993). By establishing 21 SDM properties, Karam and Casselman's cataloguing framework addresses some of these issues in a positive fashion.

Although use of the framework and AHP can be expanded to include additional SDMs, this study concentrated on two SDMs, Information Engineering (IE) and Software Engineering (SE). They are the most popular SDMs.

SE was introduced in the mid-1970s and is probably the most widely accepted SDM. Applying engineering principles, SE stresses the procedures and process required for a software system. Although also introduced in the mid-1970s, IE has only recently gained popularity. IE is a data driven SDM, focusing on the strategic, long-term goals. Identifying the data for the organization precedes designing the specific business areas.

A FRAMEWORK FOR SDM EVALUATION

Karam and Casselman provide a comprehensive framework for evaluating SDMs within an organization to enable practitioners in selecting an appropriate SDM or collection of SDMs. There are 21 properties grouped into three categories: technical, usage and managerial. There are 14 technical properties (Table 1) which encompass the techniques and processes of the SDM. The 5 usage properties (Table 2) address practical issues such as training and available tools. The 2 managerial properties (Table 3) deal with managerial issues during development. These properties are independent of any CASE tool support available for the SDM; CASE tools are only as effective as the methodologies that support them (Bloor 1993).

It is felt that this framework can be successfully applied to a variety of SDMs and it is easy to use and understand. The framework was designed to enable a practitioner to evaluate SDMs. Since IS faculty also need some guidelines when selecting a SDM, the framework was used to determine their SDM preferences.

TECHNICAL PROPERTY	DEFINITION
SYSTEM DEVELOPMENT LIFE CYCLE	Breadth of the method - different phases of development
MAJOR WORK PRODUCTS AND NOTATIONS	Usually graphical representation and documentation
PROBLEM DOMAIN ANALYSIS AND UNDERSTANDING	Tools for analyzing and understanding the problem domain - developer understanding
PHILOSOPHY	Three perspectives - structural, behavioral, functional
PROCEDURES	One per phase - guide the technical knowledge
GUIDELINES, CRITERIA, MEASURES	Advance the procedure, recognize the state of the procedure based on measures
VERIFICATION	Work products fulfill requirements at SDLC stages
FORMALITY	Precise, unambiguous mathematical definition
MAINTAINABILITY FLEXIBILITY	Ability of work product to accommodate change
REUSABILITY	Work products can be used for another project
CONCURRENT PROCESSING	Parallel event streams external to the program efficiently handled
PERFORMANCE ENGINEERING	Timely response, maximize system throughput
TRACEABILITY	Find and localize requirements in work products
METHOD SPECIFICATION	Degree to which method can be extended or specialized to a particular application

TABLE 1
SUMMARY OF TECHNICAL PROPERTIES

USAGE PROPERTIES	DEFINITION
APPLICATION AREAS	Application domains
SYSTEM SIZE	Suitable for development and maintenance
AUTOMATED SUPPORT	Software tools to support techniques
EASE OF INSTRUCTION	Effort to train a new person, availability of instructional support
MATURITY/PROJECT HISTORY	User base over time

TABLE 2
SUMMARY OF USAGE PROPERTIES

MANAGERIAL PROPERTIES	DEFINITION
SOFTWARE DEVELOPMENT ORGANIZATION	Team communications, work products, cost estimation, staffing
EASE OF INTEGRATION	Effort required to adopt a method and begin successful application

TABLE 3
SUMMARY OF MANAGERIAL PROPERTIES

THE STUDY

Faculty from several different universities completed Analytic Hierarchy Process based surveys to evaluate the SDMs. The questionnaires were generated using Expert Choice and Karam and Casselman's definitions of the 21 properties were included. The surveys were mailed to 42 IS faculty. Fourteen surveys were returned and were used in the evaluation of the SDMs.

The primary goal is the selection of the best software development methodology for an IS course or curriculum. In order to facilitate the evaluation, the 14 technical properties were grouped into 6 criteria. The groupings are:

1. Life-cycle: life-cycle, work products and notation, problem domain, procedures

2. Measures: verification, guidelines/criteria/measures, degree of formality

3. Maintenance: maintainability/flexibility, reusability, traceability, method specialization

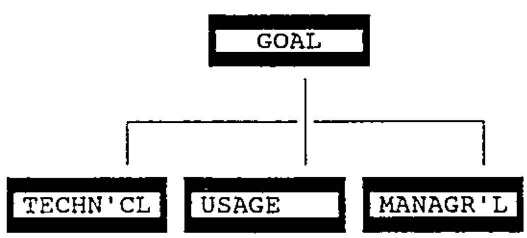
4. Performance

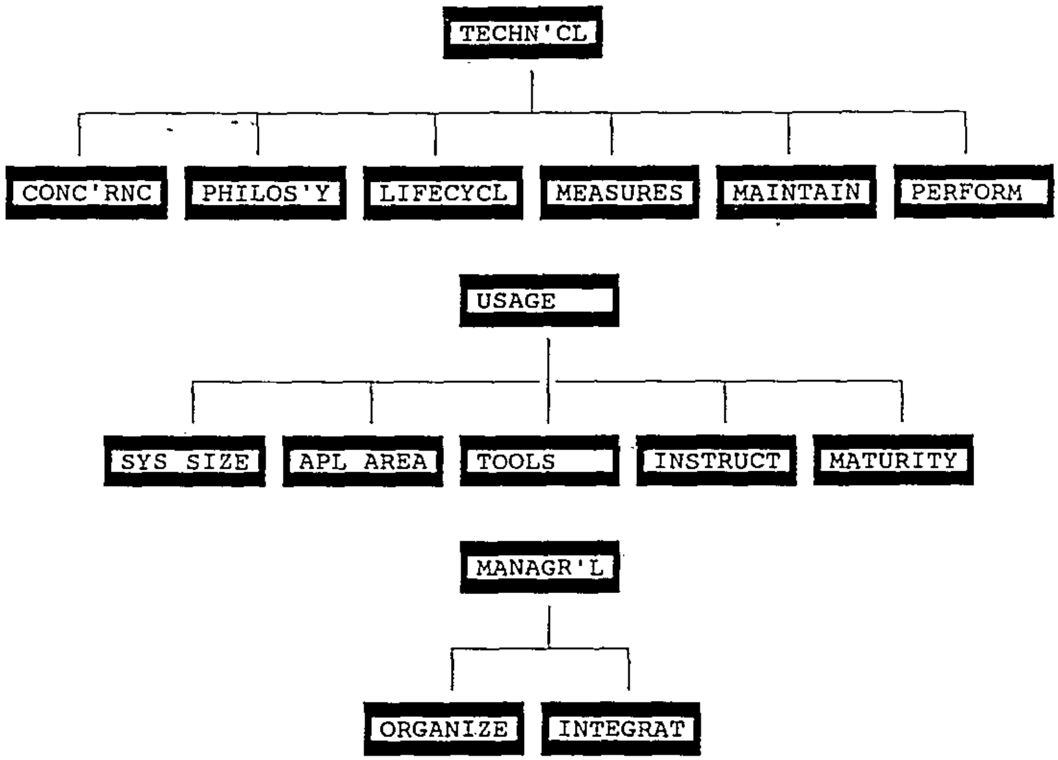
5. Philosophy

6. Concurrency.

Based on the Karam and Casselman framework and using AHP, the following hierarchies of criteria were created.

COMPARISON OF SOFTWARE DEVELOPMENT METHODS





Initially, IS faculty evaluated the relative importance of the three general properties: technical, usage and managerial. They then rated the relative importance of the 5 usage properties, 6 groupings for technical properties and 2 managerial properties. Finally, using a relative scale (Example 1) they compared the relative preference of IE and SE in an IS curriculum for each of the 13 properties.

The responses were entered into Team Expert Choice (Forman 19??). The package combined the responses and calculated aggregate values for the relative importance and relative preference measures.

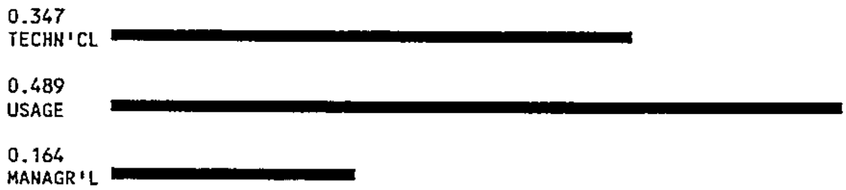
EXAMPLE 1:

Using the scale below, compare, with respect to *CONCURRENCY*, the relative **PREFERENCE** of Information Engineering and Software Engineering in an Information Systems curriculum. Someone with a perception of SE more important than IE, with a 60-40 relationship will indicate this perception as follows:

```
IE                                                              IE
0                        40                                      100
|------------------------X----------|----------------------------|
100                      60      equal                           0
SE                                                               SE
```

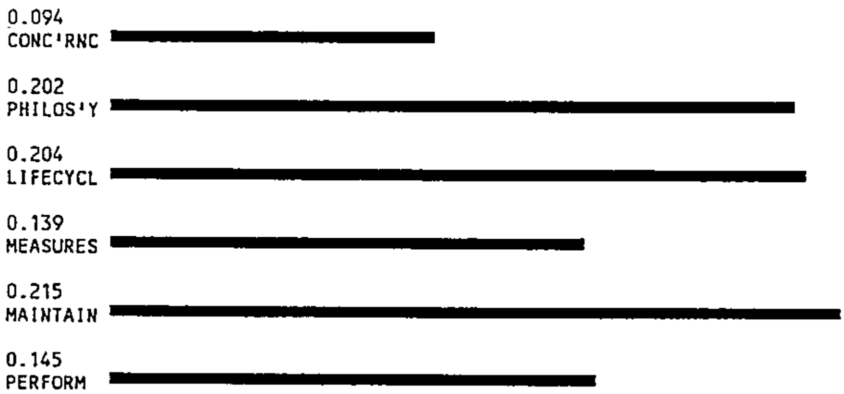
RESULTS

Priorities for the first level of analysis (Technical, Usage, Managerial) are:

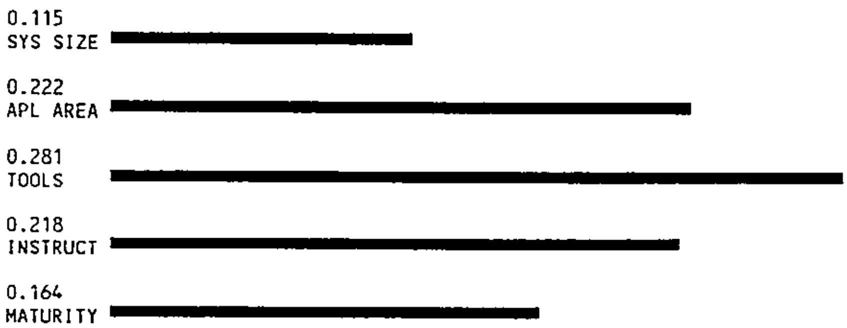


with Usage having the highest priority and Managerial the lowest. The priorities for the second level of analysis are:

Technical Properties



Usage Properties



Managerial Properties

A SDM's governing philosophy, life-cycle coverage and maintenance capabilities are the more important Technical Properties, while tool support is the more important Usage Property. The more important Managerial Property is ease of integration.

TABLE 5 contains a synopsis of the verbal judgments of importance with respect to the GOAL, Technical Properties and Usage Properties. The summary for the Managerial Properties is omitted because verbal comparisons with only 2 values can produce inaccurate results: there is no redundancy (Forman 1994). Ease of integration was 1.2 times more important than software development organization. Therefore, no one property was rated overwhelmingly more important than another.

TABLE 4 lists the preferences, given a specific SDM property, for IE and SE. IS faculty used a scale for each property, identical to Example 1, to generate the relative preferences. Overall, IE was preferred, 53.5% to 46.5%, over SE.

	SOFTWARE DEVELOPMENT METHODOLOGIES	
SDM PROPERTIES	INFORMATION ENGINEERING	SOFTWARE ENGINEERING
CONCURRENCY	0.478	0.522 *
PHILOSOPHY	0.670 *	0.330
LIFE-CYCLE COVERAGE	0.501 *	0.499
MEASURES	0.396	0.604 *
MAINTENANCE/FLEXIBILITY	0.514 *	0.486
PERFORMANCE ENGINEERING	0.351	0.648 *
SYSTEM SIZE	0.489	0.511 *
APPLICATION AREA	0.692 *	0.308
TOOL SUPPORT	0.459	0.541 *
EASE OF INSTRUCTION	0.567 *	0.433
MATURITY	0.474	0.526 *
ORGANIZATION	0.633 *	0.367
EASE OF INTEGRATION	0.561 *	0.439
OVERALL PREFERENCE	**0.535 ***	**0.465**

SDM PREFERENCES BY SDM PROPERTY
TABLE 4

* indicates preferred methodology for a given property

Verbal judgments of IMPORTANCE with respect to:
GOAL

1	TECHN'CL	9 8 7 6 5 4 3 2	█	2 3 4 5 6 7 8 9	USAGE
2	TECHN'CL	9 8 7 6 5 4 3 2	█	2 3 4 5 6 7 8 9	MANAGR'L
3	USAGE	9 8 7 6 5 4 █ 2	1	2 3 4 5 6 7 8 9	MANAGR'L

1=EQUAL 3=MODERATE 5=STRONG 7=VERY STRONG 9=EXTREME

Verbal judgments of IMPORTANCE with respect to:
TECHN'CL < GOAL

1	CONC'RNC	9 8 7 6 5 4 3 2	1	█ 3 4 5 6 7 8 9	PHILOS'Y
2	CONC'RNC	9 8 7 6 5 4 3 2	1	█ 3 4 5 6 7 8 9	LIFECYCL
3	CONC'RNC	9 8 7 6 5 4 3 2	█	2 3 4 5 6 7 8 9	MEASURES
4	CONC'RNC	9 8 7 6 5 4 3 2	1	2 █ 4 5 6 7 8 9	MAINTAIN
5	CONC'RNC	9 8 7 6 5 4 3 2	1	█ 3 4 5 6 7 8 9	PERFORM
6	PHILOS'Y	9 8 7 6 5 4 3 2	█	2 3 4 5 6 7 8 9	LIFECYCL
7	PHILOS'Y	9 8 7 6 5 4 3 2	█	2 3 4 5 6 7 8 9	MEASURES
8	PHILOS'Y	9 8 7 6 5 4 3 2	█	2 3 4 5 6 7 8 9	MAINTAIN
9	PHILOS'Y	9 8 7 6 5 4 3 2	█	2 3 4 5 6 7 8 9	PERFORM
10	LIFECYCL	9 8 7 6 5 4 3 2	█	2 3 4 5 6 7 8 9	MEASURES
11	LIFECYCL	9 8 7 6 5 4 3 2	█	2 3 4 5 6 7 8 9	MAINTAIN
12	LIFECYCL	9 8 7 6 5 4 3 2	█	2 3 4 5 6 7 8 9	PERFORM
13	MEASURES	9 8 7 6 5 4 3 2	1	█ 3 4 5 6 7 8 9	MAINTAIN
14	MEASURES	9 8 7 6 5 4 3 2	█	2 3 4 5 6 7 8 9	PERFORM
15	MAINTAIN	9 8 7 6 5 4 3 2	█	2 3 4 5 6 7 8 9	PERFORM

1=EQUAL 3=MODERATE 5=STRONG 7=VERY STRONG 9=EXTREME

Verbal judgments of IMPORTANCE with respect to:
USAGE < GOAL

1	SYS SIZE	9 8 7 6 5 4 3 2	1	█ 3 4 5 6 7 8 9	APL AREA
2	SYS SIZE	9 8 7 6 5 4 3 2	1	█ 3 4 5 6 7 8 9	TOOLS
3	SYS SIZE	9 8 7 6 5 4 3 2	█	2 3 4 5 6 7 8 9	INSTRUCT
4	SYS SIZE	9 8 7 6 5 4 3 2	1	█ 3 4 5 6 7 8 9	MATURITY
5	APL AREA	9 8 7 6 5 4 3 2	█	2 3 4 5 6 7 8 9	TOOLS
6	APL AREA	9 8 7 6 5 4 3 2	█	2 3 4 5 6 7 8 9	INSTRUCT
7	APL AREA	9 8 7 6 5 4 3 2	█	2 3 4 5 6 7 8 9	MATURITY
8	TOOLS	9 8 7 6 5 4 3 2	█	2 3 4 5 6 7 8 9	INSTRUCT
9	TOOLS	9 8 7 6 5 4 3 2	█	2 3 4 5 6 7 8 9	MATURITY
10	INSTRUCT	9 8 7 6 5 4 3 2	█	2 3 4 5 6 7 8 9	MATURITY

1=EQUAL 3=MODERATE 5=STRONG 7=VERY STRONG 9=EXTREME

TABLE 5

Also using the scale developed in Example 1, CHART 1 depicts, given an individual respondent, SDM preferences. Using the 14 criteria, 8 IS faculty preferred Information Engineering over Software Engineering.

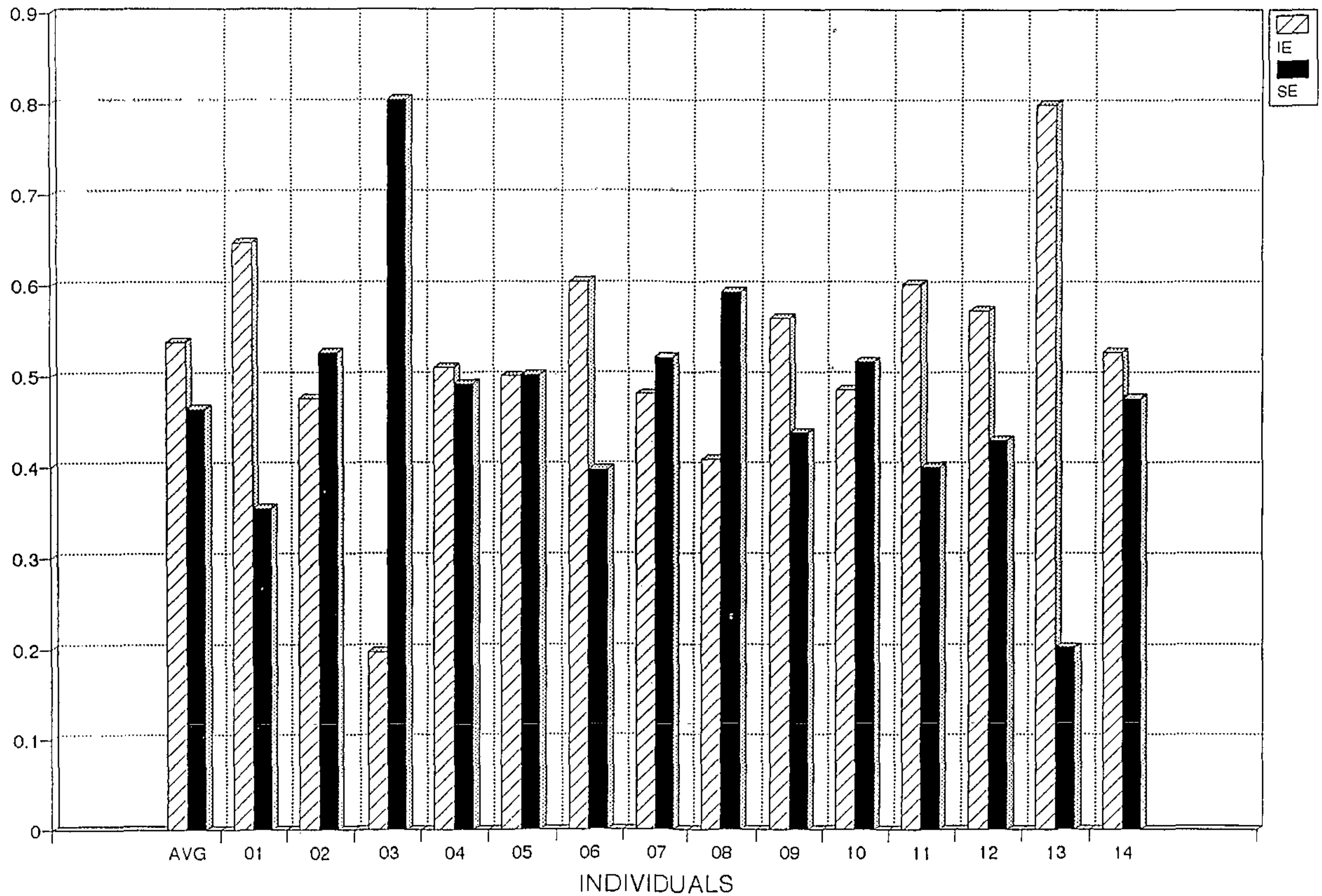
SUMMARY

The Analytic Hierarchy Process is used to convert subjective judgments of relative importance and preferences about two Software Development Methodologies into numeric values. These values are then used to rank two SDMs, Information Engineering and Software Engineering.

The criteria used are SDM properties established in Karam and Casselman's framework developed for cataloging SDMs. IS faculty from different universities used a AHP-based questionnaire to record their preferences with regard to an IS course or curriculum. They also ranked the criteria on relative importance. Their responses were combined to formulate the reported results. There was no overwhelming preference of one SDM over the other and none of the properties dominated the others.

Given the number of criteria and the subjective nature of the problem, AHP appears to be an appropriate technique for evaluation of SDMs. Future research should include using AHP and the framework with practitioners and adoption into an organizational setting. Additionally, different SDMs may be easily incorporated into the model.

CHART 1
AVERAGE AND INDIVIDUAL PREFERENCES



REFERENCES

Bloor, Robin (1993) "Stretching CASE Method" DBMS, March 1993, 12-13.

Forman, Ernest H. (1994) Team Expert Choice, Expert Choice Publishing, Pittsburgh, Pennsylvania.

Karam, Gerald M. and Casselman, Ronald S. (1993) "A Cataloging Framework for Software Development Methods." IEEE Computer, Vol. 26, No. 2, February 1993, 34-45.

Kerr, James M. (1991) "The Information Engineering Paradigm." Journal of Systems Management, Vol. 42, No. 4, April 1991, 28-32.

Keyes, Jessica (1992) "How Software is Developed Undergoing Basic Changes." Software Magazine, Vol. 12, No. 1, January 1992, 38-56.

Saaty, Thomas L. (1980) Analytic Hierarchy Process: Planning, Priority Setting, Resource Allocation, McGraw Hill, New York, 1980.

Schach, Stephen R. (1990) Software Engineering. Irwin, Homewood, Illinois.

Yau, Chuk and Davis, Tim (1993) "Using Analytic Hierarchy Process (AHP) to Prioritize Auditing Task for Large-Scale Software Systems," Journal of Systems Management, November 1993, 26-31.