Tools for Decision Analysis and Resolution

Dr. Richard D. Stutzke (late)
Science Applications International Corporation

Engineers, managers, and estimators must often select the best items from a large list. One example is which features to include in a product or system. This article describes two types of decision-making methods to help do this.

Systems integration projects require that many kinds of decisions be made, as illustrated in Table 1. The evaluation and selection of alternatives is often difficult as multiple individuals may be required to reach a consensus by considering one or more decision factors.

Making any decision requires time and money. Thus, the first decision should be whether the issue merits the use of a formal decision method. If making a wrong decision will have significant impacts for the project, then the team should use a formal decision method. Typical criteria for deciding if a decision is significant are cost, delay, safety, and corporate liability. For example, if a particular component contributes 50 percent to the total cost of the system, then careful analysis is warranted.

This article describes two types of decision-making methods: voting and multiple criteria decision-making. Voting techniques allow a group to rank alternatives based on unstated criteria. Multi-Criteria Decision-Making (MCDM) techniques allow a group of people to specifically identify important criteria and then combine ratings of these criteria to identify the *best* alternative. These methods can easily be implemented using spreadsheets.

Voting Techniques

Voting techniques provide a way to rank a set of alternatives. Political scientists and game theorists have studied many types of voting. Each voting scheme has particular advantages and disadvantages with respect to various criteria such as fairness, monotonousness, sensitivity to minor changes in the way the votes are cast, abstentions, insincerity, and susceptibility to manipulation.

Approval voting allows each voter to vote for (or approve of) as many items as he or she desires; however, each voter can cast only one vote for each item. The item with the most votes wins. (Alternatively, N items having the largest numbers of votes may win.) Approval voting is simple to understand and use. Many governments and organizations around the world use it to elect officials.

Table 2 shows an example of approval voting used to select features from a list of five features, here identified as A through E. There are five voters (stakeholders) whose votes are shown by the Xs in the table. The right-hand column shows the total votes received by each feature. Feature D is the winner, with feature B the second choice.

The bottom row of the table shows the total votes cast by each voter. The number of votes cast varies, perhaps indicating that the stakeholders did not carefully consider their choices and so the result may not truly represent the best consensus. To obtain a *better* result, use techniques that require the voters to make a *standardized commitment* and so improve the sampling of the stakeholders. Two ways to do this are the nominal group technique and multi-voting.

The *Nominal Group Technique* produces a consensus of rankings. Assume that the number of items in the list is L. Allow each person to choose N items, where N is approximately the total number of items desired on the final list. (Choosing a value of N that is less than the final number of items desired will force the stakeholders to make careful decisions.) Each person must select the N most important items, ranking them from *N* (most important) to *1* least important. (This is a good ranking system, since the items receiving no votes will have a zero score. Scoring would be more complicated if *1* denoted the most important item.) The facilitator

Table 1: *Typical Decisions for Systems Integration Projects*

- Choose product features (with/without other constraints).
- Identify the *best* design option (trade studies).
- Decide whether to make, reuse, or buy.
- Select a Commercial-off-the-Shelf (COTS) component or tool.
- Pick a vendor or subcontractor.
- Select a risk mitigation approach.
- Decide to bid or not to bid.
- Terminate integration testing.
- Modify work products that are already baselined.

Table 2: *Approval Voting Example*

Feature	Stakeholders					Total Votes
	1	2	3	4	5	
A		X				1
B		X	X		X	3
C						0
D	X	X	X	X	X	5
E	X					1
Total Votes By Voter	2	3	2	1	2	

Table 3: *Nominal Group Technique Example*

Feature	Stakeholder					Total Votes
	1	2	3	4	5	
A		1	2		2	5
B	3	2	3	3	3	14
C						0
D	2	3	1	2	1	9
E	1			1		2

Feature	Stakeholder						Total	Rank
	1	2	3	4	5	6	Votes	
A							0	-
B	3	1	1	1	1		7	1
C			1	1		1	3	3
D		1			2		3	4
E		1	1	1		1	4	2
F							0	-
G						1	1	5
H							0	-
Total Votes Cast	3	3	3	3	3	3		

Table 4: *Example of Multivoting*

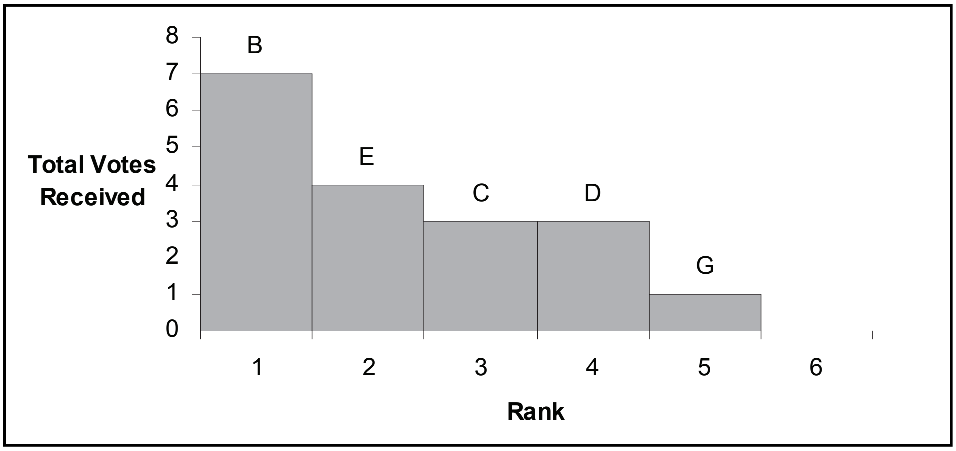


Figure 1: *Pareto Ranking of the Scores*

Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree

Never	Seldom	Sometimes	Often	Always

Hourly	Daily	Weekly	Monthly	Yearly

Table 5: *Some Preference Ratings (Likert Scales)* [1]

collects the votes and totals the votes for each item. The item with the largest total wins.

Table 3 (see previous page) shows an example of five features (A through E) with five stakeholders (1 through 5). Each stakeholder was allowed to choose and rank the three most important items. The last column shows the total score. Feature B is best, with Feature D ranked second.

In *multi-voting*, each stakeholder has a certain number of votes, N, to cast as he or she likes. A stakeholder can cast one, two, or even all N votes for a single item. Table 4 shows an example of eight features (A through H) and six voters (stakeholders). Each voter is given three votes to cast. The table shows the total votes for each feature and ranks the features based on these totals. Feature B is ranked first. Feature E is ranked second. Features C and D have the same total score, but Feature C was selected for third place because three stakeholders voted for it while only two voted for Feature D.

Table 6: *Some Preference Ratings (Likert Scales)*

Very Low	≤ four months
Low	> four months and ≤ one year
Nominal	> one year and ≤ three years
High	> three years and ≤ six years
Very High	> six years

Pareto histograms help one to visualize the strength of the stakeholders' preferences. Figure 1 shows the scores for each item arranged in descending order. Feature B is clearly preferred. Features E, C, and D are close in value. A change of one vote from E to C (or D) would cause C (or D) to move into second place. Feature G is a distant fifth and is only one vote away from obscurity (with Features A, F, and H).

These methods can be used iteratively. The group can vote on a set of alternatives and then eliminate the items having the fewest votes. Optionally, they can discuss the top few items and revise the item descriptions. Then they repeat the process with the revised list.

Techniques for Handling Multiple Criteria

Estimation often involves ranking objects based upon *multiple* criteria. For example, due to limited resources, one might need to choose which set of tasks is the most important to implement, or which design is the best among a set of possible designs.

The stakeholders may also be involved in determining the value or utility of a particular product or process. Often, the value, or benefit, depends on many subjective factors that are difficult to quantify. The academic discipline of decision-making deals with models (normative models) that help associate and measure such factors. The two main types of models are *expected utility theory* and *multi-attribute utility theory* (MAUT).

In *expected utility theory*, the value of a particular outcome is determined by estimating the probability of that outcome and multiplying it by the estimated utility of that outcome. The probability and utility are expressed as real numbers. For example, suppose that a lottery ticket is purchased for $5.00. If the ticket is a winner, the ticket holder will receive $10,000. If the chance of winning the lottery is one in one million, however, the expected return is only $0.01 (= 10^{-6} x 10^4). In this case, the expected return does not justify the price of the ticket. Quantitative risk assessment is based on utility theory. Specifically, the risk impact equals the probability of occurrence times the cost of occurrence.

MAUT extends expected utility theory to decisions with multiple alternatives that depend on many attributes or criteria. MAUT maximizes a function of the various criteria, and it assumes that one criterion can counterbalance another (substitution). For example, cost reductions or revenue increases can offset investment or implementation costs. To define a MAUT technique, consider two things: how the properties map to an appropriate measurement scale that preserves the relations between the objects[1] or how orthogonal properties (those with no overlap in what each measures) are selected, and how to define the particular function to maximize. A related issue is defining how to estimate the function's parameters (e.g., the weights that are used to combine ratings for different criteria).

MAUT techniques use preference ratings, illustrated in Table 5, which are ordinal measurements. The ordinal scale only allows comparisons and equality. A restricted ordinal scale that places objects into bins may also be defined. Table 6 illustrates such a scheme. Many parametric estimation models use such rating scales.

To use MAUT do the following:

1. Determine a measurement scale for each attribute, R_i.
2. Map the rating, R_i, to a (ratio scale) value, V_i.
3. Specify the relative weights of the attributes, W_i.

Then compute the score using the weighted sum of the values[2]. For example, suppose that one wants to evaluate four COTS components, named A, B, C, and D. (This example comes from Section 27.5 in [2].) One will want to consider the following factors:

1. Functionality.
2. Integration Effort (person-weeks).
3. Cost (dollars).
4. Vendor Reputation.
5. Product Maturity.
6. Developer Toolkit Available.
7. Training Availability.

The top half of Table 7 lists the rating scale for each factor and its relative weight. (The weights sum to 1.0, but this is not necessary.) The bottom half of Table 7 shows the (ratio scale) values assigned to each rating. Table 8 shows the evaluator's ratings for each factor for each component, with the total weighted score shown in row 6. Component D is the best, with Components A and C nearly tied for second place.

The *Analytic Hierarchy Process (AHP)* is a popular MAUT technique developed by Thomas Saaty [3]. AHP addresses multiple criteria, including subjective criteria. AHP constructs a multi-level hierarchy, such as the one shown in Figure 2. The top level is the decision objective. The bottom level (the leaves) is the possible actions or alternatives. The intermediate levels represent factors that affect the preference or desirability of one alternative, or subfactors that contribute to a factor.

The steps of the AHP process are the following:

1. Define the Problem.
2. Construct the Hierarchy.
3. Establish Element Comparisons.
4. Calculate Element Priorities.
5. Calculate Overall Priorities.

The decision-maker identifies which factors are important and defines how they influence one another. Then evaluators make pair-wise comparisons at each level, capturing these in *judgment matrices*, one for each criterion or alternative. The rating matrix is triangular since complementarities are assumed. That is, if A >> B then B << A. If A_{ij} denotes the rating of object i relative to object j for attribute A, the AHP model assumes that: $A_{ji} = 1/A_{ij}$. Since an object is identical (equal) to itself, $A_{ii} = 1$. For a single criterion having N alternatives, the evaluator must make N*(N – 1)/2 comparisons. Saaty uses a nine-value preference scale. (Other rating scales can be used if desired.)

AHP uses matrix calculations to determine the preference ratios for

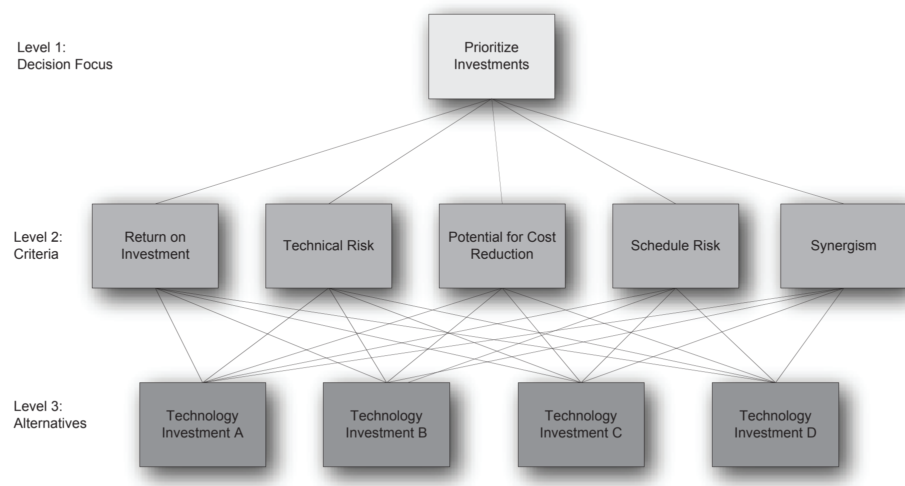
	G	H	I	J	K	L	M
7	Factor Name	Weight	Ratings				
8	Functionality	0.30	VL	LO	NM	HI	VH
9	Integration Effort (person-weeks)	0.20	1	2	3	4	5
10	Cost (dollars)	0.20	below 500	500 to 1000	above 1000		
11	Vendor Reputation	0.10	Poor	Unknown	Good	Excellent	
12	Product Maturity	0.10	VL	LO	NM	HI	VH
13	Developer Toolkit available	0.05	No	Yes			
14	Training availability	0.05	No	Some	Full		
15		0.00					
16							
17	Factor Name		Values				
18	Functionality		1.00	2.00	3.00	4.00	5.00
19	Integration Effort (person-weeks)		1.00	2.00	3.00	4.00	5.00
20	Cost (dollars)		1.00	0.00	-1.00		
21	Vendor Reputation		-5.00	-3.00	0.00	9.00	
22	Product Maturity		1.00	2.00	3.00	4.00	5.00
23	Developer Toolkit available		0.00	1.00			
24	Training availability		1.00	2.00	3.00		
25							

Table 7: *User-Specified Criteria and Weights*

	A	B	C	D	E
1	**Description:**	Evaluation of Four COTS products			
2	**Prepared By:**	Mary Smith			
3	**Date Prepared:**	14-Oct-04			
4					
5	**Option =**	**A**	**B**	**C**	**D**
6	**Score =**	**1.80**	**1.30**	**1.75**	**2.30**
7	**Factor**				
8	Functionality	HI	NM	LO	VH
9	Integration Effort (person-weeks)	1	2	3	4
10	Cost (dollars)	500 to 1000	below 500	500 to 1000	above 1000
11	Vendor Reputation	Good	Poor	Good	Unknown
12	Product Maturity	NM	LO	HI	NM
13	Developer Toolkit available	No	Yes	No	Yes
14	Training availability	Some	No	Full	Full
15					

Table 8: *Weighted Scores for the COTS Components*

Figure 2: *A Hierarchy for Technology Investment*



objects and the importance ratios (relative weights) for the criteria at one level (expressed as a priority vector). There are several ways to calculate the priority vector. Saaty computes matrix eigenvalues. Eduardo Miranda uses a geometric mean for a single criterion (software size) in [4]. The calculations in Step 5 usually include consistency tests, allowing planners to revise their relative comparisons to be more realistic (i.e., they repeat Step 4). Then the AHP model combines the relative weights to obtain a ranked list of alternatives (a ratio scale preference vector).

Summary

This article describes simple techniques for ranking and selecting items, as well as ways to evaluate alternatives based on multiple attributes. These techniques all assume rational behavior on the part of the participants. Challenges in applying such techniques in practice are the following:

- People do not always make perfect decisions (due to ignorance, biases, or manipulative strategies).
- People may change their minds.
- There may not be enough resources (time, money) to assign good ratings to all the factors identified.
- Key factors that greatly affect the desirability of the alternatives may not be identified (e.g., the unexpected discovery of an endangered species residing on the project construction site).
- It may be difficult to identify orthogonal criteria.◆

Recommended Reading

For additional information about all of these techniques, including many references, see Chapter 27 of [2].

Part III of [5] is a good place to start when evaluating the economic value of products and systems. Part IIIA deals with cost-effectiveness analysis. Part IIIB discusses multiple-goal decision analysis. Part IIIC discusses uncertainties, risk, and the value of information. Boehm specifically addresses quantities of interest to software and system engineering.

The business community has extensively studied decisions involving financial analysis. Two references that address software-related business decisions are “Return on Software: Maximizing the Return on Your Software Investment” by Steve Tockey [6] and “Making a Business Case: Improvement by the Numbers” by Donald J. Reifer [7].

References

1. Likert, Rensis. “A Technique for the Measurement of Attitudes.” Archives of Psychology 140 (1932).
2. Stutzke, Richard D. “Estimating Software-Intensive Systems.” Addison-Wesley, 2005.
3. Saaty, Thomas. The Analytic Hierarchy Process. New York: McGraw Hill, 1980.
4. Miranda, Eduardo. “Improving Subjective Estimates Using Paired Comparisons.” IEEE Software 18.1 (2001): 87-91.
5. Boehm, Barry W. “Software Engineering Economics.” Prentice-Hall, 1981.
6. Tockey, Steve. “Return on Software: Maximizing the Return on Your Software Investment.” Addison-Wesley, 2005.
7. Reifer, Donald J. “Making a Business Case: Improvement by the Numbers.” Addison-Wesley, 2001.

Notes

1. Measurement assigns directly observed (or estimated) values of some attribute to a mathematical representation that preserves the relationships between the objects in the real world. This guarantees that the mathematical objects can be manipulated and valid conclusions about the corresponding real-world objects drawn. A measurement scale defines a representation and a set of allowed operations on the objects.
2. This is a linear function. This is the usual approach, but the MAUT technique also works with a non-linear objective function.

About the Author

Richard D. Stutzke, Ph.D., was an employee of Science Applications International Corporation (SAIC) and had more than 40 years of experience with software development and project management for scientific, embedded real-time, and commercial systems. Stutzke authored more than 50 papers and articles on software development, estimation, metrics, and management. Stutzke was the principal author of SAIC’s software estimating courses and taught more than 1,000 students since developing the course in 1990. His book, “Estimating Software-Intensive Systems: Projects, Products and Processes,” won the SAIC Executive Science and Technology Council 2005 Publication Prize Contest in the Technical Book category. The selection criteria included originality of work, significance of results, and effectiveness of presentation. Chapter 27 covers many topics of interest to project planners, systems engineers, and software engineers, including the topics described in this article. For a description of his book, see the book’s Web page <http://sw-estimation.com>.

Letter to the Editor

Dear CrossTalk Editor,

I very much liked the theme of your August 2007 issue *Stories of Change* and the change articles. Great topic and one I don’t recall seeing before. It was a good reminder that implementing change is not simple but it can be managed successfully. I particularly liked the article *Good News From Iraq*. If change can be successful in that environment, what excuse can the rest of us have for failing?

I would like to add one point about resistance to change, and that is trying to do too much change at one time. People can simply be on overload even if they desire the change. Case in point: In the mid-90s, my base was undergoing a base realignment and closure process, we did a reorganization to a matrix-type organization, and we were downsizing. Three big changes all at once! Any one of them by itself would have been a challenge. A word to the wise: Too much change at once is too confusing, especially if your people are already on overload.

Keep those CrossTalk issues coming!

– Alan Kaniss
United States Navy
<alan.kaniss@navy.mil>