

Toward an assessment of software development risk

Barki, Henri; Rivard, Suzanne; Talbot, Jean . Journal of Management Information Systems : JMIS; Armonk Vol. 10, Ed. 2, (Fall 1993): 203.

[Link para o documento do ProQuest](#)

RESUMO

Despite the introduction and use of a wide variety of system development methods and tools, software projects are still plagued by time and cost overruns and unmet user requirements. To avoid these problems, it is frequently recommended that the risk associated with a software project be managed. A task that is critical to the proper management of software development risk is the assessment of the risks facing the project. Based on previous research, a definition and a measure of software development risk are proposed. Subsequently, data collected in a survey of 120 projects is used to assess the reliability and validity of this instrument.

TEXTO COMPLETO

A SAMPLING OF "RUNAWAY" PROJECTS

ALLSTATE INSURANCE: In 1982, with software from Electronic Data Systems, the insurer began to build an \$8 million computer system that would automate it from top to bottom. Completion date: 1987. An assortment of problems developed, delaying completion until 1993. The new estimated price: \$100 million.

BUSINESS MEN'S INSURANCE: In 1985 the reinsurer began a one-year project to build a \$500,000 system to help minimize the risk of buying insurance policies held by major insurers. The company has spent nearly \$2 million to date on the project, which is in disarray. The new completion date is early 1990 [32, p. 165].

Software development problems from the 1970s? No. As the dates indicate, these horror stories are taking place even now in the 1990s. Unfortunately, it appears that software development efforts still suffer from age-old difficulties of cost overruns, project delays, and unmet user needs. And this, despite the recent introduction and widespread use of a plethora of approaches, techniques, and tools such as prototyping, data modeling, structured methods, fourth-generation languages, relational DBMSs, object-oriented programming, and computer assisted software engineering (CASE). Interestingly, a well-known article by McFarlan [26] on software risk management also began, one decade ago, with development failure examples just like the ones listed above. According to McFarlan, failure to assess individual project risk and to adapt management methods accordingly was a major source of the software problem. Even though McFarlan's views have been well received by practitioners and researchers alike, little research has since been done to advance our knowledge of software development risk. An important step in advancing our knowledge on this subject would be the development of a sound definition and a measure for the construct of software development risk. Such a measure would enable researchers and practitioners to quantify risk, which is "One of the most dominating steps in the risk assessment process" [21, p. 170]. While examined by some authors [2, 16, 26], the concept of software development risk in the IS domain still lacks a generally accepted means of assessment. In this paper, we attempt to address these issues by developing a conceptual definition and proposing an initial measure for this construct. We also provide a preliminary evaluation of our measure via data collected in a field study of 120 software development projects.

THE CONCEPT OF SOFTWARE DEVELOPMENT RISK

The term risk is associated with many human endeavors be it space exploration, nuclear reactor construction, company acquisition, security evaluations of information systems, or information systems development. As such, people in a variety of domains have studied the notion of risk. Even though different approaches or perspectives to studying risk exist, an examination of the literature in various domains reveals a great degree of similarity. In essence, many definitions of risk comprise two dimensions: (1) the probability associated with an undesirable event, and (2) the consequences (usually financial) of the occurrence of this event. Speaking of total risk management, Haimes [21] states, "Risk is often defined as a measure of probability and severity of adverse effects" [p. 169]. In the field of engineering, risk is defined as "a combination of the probability of an undesirable event with the magnitude of each and every foreseeable consequence (damage to property, loss of money, injury to people, lives lost, and so on)" [9, p. 24]. Along the same vein, project risk in project management is defined as the degree of exposure to negative events and their probable consequences [38]. Similarly, in measuring the risk of software failure during its operation, Sherer combines estimates of loss magnitudes with failure probabilities [33]. In the management of software projects Charette [16] states that for an event or action to be considered a risk, there must be a loss associated with it, chance, and some choice involved. Finally, with reference to software risk management, Boehm [13] defines risk impact or risk exposure (RE) as

$$RE = \text{Prob(UO)} * \text{Loss(UO)}$$

where Prob(UO) is the probability of an unsatisfactory outcome, and Loss(UO) is the loss to the parties affected if the outcome is unsatisfactory. [p. 4]

Thus, to assess the degree of risk it would be necessary to determine probabilities of undesirable events and their associated losses. In several contexts, risk assessment is performed by evaluating both dimensions quantitatively. In reviewing quantitative risk analysis methodologies, Rainer, Snyder, and Carr [29] describe these methods as "based on regarding loss exposure as a function of the vulnerability of an asset to a threat multiplied by the probability of the threat becoming a reality" (p. 133). For example, a frequently used approach, Probabilistic Risk Assessment, provides "estimates of overall accident probabilities synthesized from the design and performance characteristics of individual components of the technology such as pipes, pumps, valves, pressure vessels, control equipment, and human operators" [25, p. 239]. Livermore Risk Analysis Methodology [20], and Stochastic Dominance [28] are other examples of quantitative risk analysis methodologies.

Unfortunately, assessing risk via a quantitative evaluation of probabilities has two shortcomings. First, in many situations probability distributions of undesirable events are very difficult and unreliable to estimate [28, 34]. These estimation problems are present in many domains including RD, engineering, insurance, and business policy, as well as IS. Second, many authors draw attention to the relative nature of risk, pointing out that absolute risk does not exist, and that "It is a subjective thing—it depends upon who is looking" [23, p. 12]. Proponents of this view argue that risk involves uncertainty and loss rather than probabilities and loss. Referring to risk in its general sense, Kaplan and Garrick [23] state: "The notion of risk, therefore, involves both uncertainty and some kind of loss or damage that might be received" [p. 12]. Their point is succinctly illustrated by Denenberg et al. [18], who state: "Columbus believed that the world was round and that he could reach the Orient by sailing westward. Many of his contemporaries, however, believed that the world was flat and that Columbus' ships would fall off the edge if he sailed too far west. Uncertainty of loss existed when Columbus set sail, even though we now know that there was no possibility of that particular loss" [p. 5].

In lieu of estimating probabilities of undesirable events in assessing risk, alternative risk evaluation methods identify and assess factors that influence the occurrence of these events [4, 5, 14, 22, 35, 38]. For instance, Anderson and Narasimhan [5] propose an approach whereby risk factors are identified and used as independent variables in a discriminant function. The dependent variable is the discriminant score, and "in essence, capsulizes the effect of all the risk factors" [p. 5 14]. Arguing that many people have little knowledge about probabilistic concepts, and that therefore they cannot express themselves in the mathematical terms required for a probabilistic assessment of negative effects, Kangari and Boyer [22] adopt a risk assessment method based on the use of natural language. With this method, the uncertainty factors are first identified. Then, the people involved are asked to express, in natural language, the relative weight of and severity of loss due to each factor. The factor

weights are subsequently associated with membership values in their respective fuzzy sets which are then combined into a single fuzzy set in order to assess overall risk.

In light of the above discussion, it would seem appropriate to define project development risk by referring to the uncertainty surrounding a project and the magnitude of potential loss associated with project failure. Thus, we define

Software development risk = (project uncertainty) * (magnitude of potential loss due to project failure).

This definition, consistent with Kaplan and Garrick [23] and Denenberg et al.'s [18] recommendations, differs from Boehm's in two respects. First, the above definition refers to uncertainty rather than to probability. Second, while Boehm identifies several potential unsatisfactory outcomes for a given project, our definition assumes a single unsatisfactory outcome: project failure.

A number of IS researchers have identified factors threatening successful software development. For example, after studying the implementation of 56 decision support systems, Alter [2] identified eight such factors: nonexistent or unwilling users; multiple users or implementors; turnover among all parties; inability to specify purpose or usage; inability to cushion impact on others; loss or lack of support; lack of experience; and technical or cost-effectiveness problems. Reflecting on the management of software development projects, McFarlan [26] pointed out and operationalized three dimensions influencing the risk inherent in a project: project size; experience with the technology; and project structure. Recognizing the difficulties associated with making accurate estimates of probabilities and losses related to software development, Boehm [13] recommended the use of approximate methods, and proposed a prioritized checklist of ten software risk items: personnel shortfalls; unrealistic schedules and budgets; developing the wrong software functions; developing the wrong user interface; gold plating; continuing stream of requirement changes; shortfalls in externally furnished components; shortfalls in externally performed tasks; real-time performance shortfalls; and straining computer science capabilities. Under the heading of uncertainty factors, the IS literature also refers to factors that influence the outcome of a software development project. For example, Zmud [39] states that technological complexity, the degree of novelty or structure of the application, technological change, and project size influence the outcome of large software development efforts. Discussing the role of uncertainty in determining user information requirements, Davis [17] identifies four sources of project uncertainty: the task to be supported; the application to be developed; the users; and the analysts. Beath [8] suggests that several sources of uncertainty be taken into account in the management of software development projects: complexity, lack of structure, or instability of project objectives; newness of the technology; users; IS management; upper management; and project size.

Our review of the literature reveals a high degree of resemblance between what some authors have labeled "risk factors" and what others have called "uncertainty factors" in IS. For example, team turnover, a risk factor suggested by Alter [2], and personnel changes, an uncertainty factor identified by Zmud [39], share the same meaning. Similarly, a risk factor identified by Anderson and Narasimhan [5] is lack of top management involvement, while an uncertainty factor suggested by Beath [8] is lack of upper management support. Given these similarities, the lists of uncertainty and risk factors identified in the IS literature were examined in order to group factors with shared meanings. The concept underlying each group of factors was then identified. The results of this exercise are presented in Table 1. (Table 1 omitted).

As indicated by the column entitled underlying concept in Table 1, the risk and uncertainty factors, as defined in the IS literature, both appear to relate to the same construct. In other words, all of these factors can be viewed as influencing the amount of information required to manage a project. Since the amount of information needed to accomplish a task is generally used as the basis for defining task uncertainty [19], we would argue that risk and uncertainty factors, as discussed in the IS literature, are one and the same, and should all be named uncertainty factors. This recommendation is also consistent with the software development risk definition proposed above.

DEVELOPMENT AND ASSESSMENT OF A SOFTWARE RISK MEASURE

INSTRUMENT DEVELOPMENT

A comprehensive review of the IS uncertainty/risk literature was conducted. This review resulted in the

identification of 35 risk variables, listed in Table 2 (34 variables related to uncertainty, and one related to magnitude of potential loss). Next, a questionnaire operationalizing the 35 variables was developed by using existing scales when available [1, 7, 26, 30, 31, 36, 37], and by developing new scales. The resulting questionnaire, measuring the uncertainty and the magnitude of potential loss related to a system development project, contained 144 items.

TABLE 2 SOFTWARE DEVELOPMENT RISK VARIABLES

PROJECT UNCERTAINTY--SCALE SOURCE

Need for new hardware--[26]
 Need for new software--[26]
 Number of hardware suppliers--[26]
 Number of software suppliers--[26]
 Number of users outside the organization--[26]
 Number of departments
 Degree of computerization of current system
 Number of people of team
 Relative project size
 Team diversity
 Number of users in the organization
 Number of hierarchical levels occupied by users
 Lack of development expertise in team--[1]
 Team's lack of expertise with application--[26]
 Team's lack of expertise with task--[1]
 Team's lack of general expertise--[1]
 Number of similar projects leader managed
 Leader lack of familiarity with team
 Dependence on a few key people
 Lack of user experience and support--[7, 26]
 Project leader's experience
 Technical complexity
 Number of links to existing systems
 Number of links to future systems
 Extent of linkage to other organizations
 Extent of changes
 Resource insufficiency
 Intensity of conflicts--[1, 31]
 Lack of clarity of role definitions
 Task complexity--[36, 37]
 Top management support
 Quality of software supplier support
 Quality of hardware supplier support
 Extent of changes in the project
 Magnitude of potential loss

As can be seen from Table 2, the 35 variables, and therefore the 144 items measuring them, pertain to various characteristics of a software development project. Since these characteristics are related to technical aspects of a project as well as to its user environment, their reliable assessment requires that two sources be used: the project leader, and the future users of the system. Consequently, two questionnaires were designed: one for the project leader, and one for a representative user. A pretest of both questionnaires was subsequently conducted with ten

project managers and eight users, leading to minor modifications. The modified questionnaires were then used in a large-scale survey to purify the scales they contained.

DATA COLLECTION

To obtain the study sample, IS managers of the largest 100 companies in Quebec, and all the ministries, government agencies, and public corporations in the province were first sent a letter describing the study. Two to three weeks following the mailing of a letter, recipients were contacted by phone and asked if their organization had projects that would qualify for our study, and if they were willing to participate. In order to be included in the sample, a project had to be ongoing, with system conversion not yet completed.

The purpose of this requirement was twofold. The first objective was to avoid retrospective bias. The second objective was to create a sample of projects at varying stages of completion. Given that project risk exists throughout the life of a project and that it should be assessed and managed on an ongoing basis [10, 38], such a sample would be more representative of the phenomenon under study. The final sample consisted of project leaders and user representatives from 120 ongoing projects in 75 organizations.

Two respondents were identified for each project: the project leader and a user representative. The user representative was a key informant selected via interviews with the IS manager and the project leader. Selection criteria included knowledge of the user community, project objectives, and organizational goals. A structured interview with each respondent was conducted, and they were each given a questionnaire for later collection. The purpose of the structured interview was to gather factual and descriptive information about the project. Tables 3 and 4 describe various sample characteristics. (Tables 3 and 4 omitted). As can be seen, the sample contains organizations from a variety of economic sectors, of differing size, and with a wide range of employees in the IS departments. The sample even includes one organization with no IS department with the project being developed by outside consultants. Since the project was considered important for the organization, assessing its risk was considered relevant warranting its inclusion in the sample.

DATA ANALYSIS

To assess the reliability and the validity of the risk measure, operationalized with 35 variables, a three-step procedure was followed. In the first step, multiple-item scales were examined via Cronbach alpha and factor analysis. This analysis resulted in the elimination of two scales that had undesirable psychometric properties, leaving 33 variables (of which twelve are multiple-item scales).

For each multiple-item scale, items loading less than 0.5 to a factor were discarded. In addition, an examination of the Cronbach alpha values for each variable led to the retention of a final set of items. The final reliability scores of these twelve multiple-item scales are presented in Table 5. (Table 5 omitted). As can be seen, nine variables have Cronbach alphas above 0.80. The remaining three have alphas of 0.68, 0.70, and 0.73. These values were deemed adequate for the purposes of this study, and the twelve variables retained for further analysis.

The means and standard deviations of these twelve multiple-item scales, and the remaining 21 single-item variables (of which eleven are ratio scales), were calculated. Table 6 contains the means of the eleven ratio-scale variables. (Table 6 omitted). Table 7 lists the standardized means of the 22 non-ratio-scale variables. (Table 7 omitted). An examination of the two tables indicates that the sample of 120 projects contains substantial variation. Moreover, most standardized non-ratio-scale means are close to or slightly above 0.5.

In the second step, the 32 uncertainty variables were factor analyzed in order to examine the underlying structure of the uncertainty construct. The initial solution contained nine factors with eigenvalues above 1.0 and the proportion of the variance explained by the nine factors was 65.9 percent. The nine factors obtained were hard to interpret. In addition, they did not map to the three factors [26], four factors [17], and five factors [12] underlying the concept of the uncertainty of a software development project. Such situations, that is, where "the number of factors retained tends to be much larger than the number of factors that the researcher is willing to accept...[have] forced researchers to apply another criterion—that of substantive significance, which is applied after finding statistical significance" [24, p. 42]. Consequently, factor solutions with 3, 4, and 5 factors were also analyzed. An examination of the variables loading into the different solutions led to the selection of the 5 factor result as the

best interpretable solution. The variables loading into each of the five factors are presented in Table 8. (Table 8 omitted). Nine of the 32 uncertainty variables are excluded from the five factors because they either had factor loadings inferior to 0.4 or because they loaded significantly into more than one factor. Of the 23 uncertainty variables retained, only task complexity has a factor loading less than 0.4. In view of the fact that its loadings into other factors were near zero, we decided to keep this variable. The final list of 24 variables (23 uncertainty variables and the magnitude of loss variable) and the items with which they were measured are described in the appendix.

The interpretation of the five retained factors is relatively straightforward. The five variables loading into factor 1 are all related to the novelty aspects of a project. The quantity of new hardware to be acquired and software newness are clearly related to novelty. In addition, the larger the number of hardware and software suppliers of a project, the more likely the project team will have to deal with new hardware and software. Finally, the larger the number of users outside the organization, the more likely that new requirements will have to be taken into account. Given that four of the five variables loading into factor 1 were related to technology, this factor was named technological newness.

The five variables of factor 2, number of people on the team, project size, the team's diversity, number of future users, and the number of hierarchical levels occupied by the future users, all pertain to the size or scope of an application. Thus, this factor was named application size.

Four of the five variables loading into factor 3 are about the team's expertise, while the fifth variable concerns the users' experience and their level of support for the project. Consequently, this factor was named lack of expertise. The three variables of factor 4 are technical complexity, and the number of links the application has to existing and future systems. Since the number of links to other systems is related to complexity, this factor was named application complexity.

The fifth factor was named organizational environment. It includes variables that relate the application or the team to the organization. For example, the extent of changes relates the organization to the application because it deals with the changes that the application will bring to users' tasks. Similarly, resource insufficiency is concerned with the lack of resources provided to the project by the organization, thereby linking the application and the organization. One variable that links the team to the organization is intensity of conflicts, since it deals with conflicts between team members and users.

In the third step of the analysis, relationships between criterion variables and overall uncertainty, magnitude of loss, overall risk, and factor scores were examined. Measures for criterion variables were obtained for each project during the initial interview with the project leader who was asked to respond to eight low/high semantic differential items. Three items asked the respondent to evaluate the project's overall degree of risk, overall degree of uncertainty, and overall degree of magnitude of potential loss, respectively. The remaining five items asked the project leader to evaluate the extent to which each of five uncertainty dimensions contributed to the riskiness of the project. The five dimensions were the uncertainty related to (1) the characteristics of the application itself, (2) the future users of the system being developed, (3) the development team, (4) the task(s) being computerized, and (5) the characteristics of the organization. Since, a priori, we did not know what uncertainty dimensions would emerge in the study, the five criterion variables were based on four dimensions identified by Davis [17], and a fifth dimension (characteristics of the organization) drawn from Zmud [39], Beath [8], and Bernier [11]. The project leader's responses, on a 1-7 scale, to the eight items constitute the measures of the eight criterion variables. The analysis proceeded by an examination of the canonical correlations between the five uncertainty criterion variables and the five factors derived in the study. The first two canonical variates obtained were significant and had eigenvalues of 0.67, and 0.23, with corresponding canonical correlations of 0.63, and 0.43 respectively, indicating that linear composites of the criterion variables correlate highly and significantly with linear composites of the five derived factors. The criterion variables that correlated significantly with the first canonical variate (of criterion variables) were the risk due to the uncertainties related to the development team ($r = 0.71$), the application ($r = 0.71$), the task ($r = 0.52$), and the users ($r = 0.47$). The derived factors that correlated significantly with the first

canonical variate (of derived factors) were lack of expertise ($r = 0.83$), technological newness ($r = 0.34$), and application size ($r = 0.30$). Variables correlating highly with the second canonical covariate were the risk related to organizational uncertainty ($r = 0.94$) for criterion variables, and organizational environment ($r = 0.86$) and application complexity ($r = 0.43$) for the derived factors. These results indicate that all five derived factors are related to the five criterion variables, thus providing evidence of criterion validity.

Second, by taking the five uncertainty criterion variables and the magnitude of loss criterion variable as dependent variables (one at a time), six regressions were run with the five derived uncertainty factors and the magnitude of loss scale as independent variables. An examination of the factors that enter each regression with significant beta weights (Table 9) provides further evidence of criterion and content validity for the factors derived in this study. (Table 9 omitted). As can be seen, when the dependent variable was project risk due to the uncertainties related to the development team, the factor lack of expertise was the only significant one to enter the regression. With magnitude of potential loss as the dependent variable, the factors with significant beta weights were magnitude of loss and application size. In the case of risk due to uncertainty related to organizational characteristics, the factors entering were organizational environment and magnitude of loss. When the dependent variable was risk due to the uncertainty related to the task being computerized, the significant factors were organizational environment, which incorporates task complexity, and lack of expertise. When the application's characteristics was the dependent variable, factors entering the regression equation with significant beta weights were technological newness (a characteristic of the application) and lack of expertise. Finally, for risk due to the future users of the application, the significant factors were application size (which incorporates number of users, and their hierarchical levels) and expertise (which incorporates user experience and support). Thus, four of the five uncertainty factors (application complexity being the exception) and the magnitude of loss scale explain significant portions of the variance in their respective or related criterion variables, thus providing further evidence of content and criterion validity. Finally, correlations between the criterion variable of overall risk, and the calculated scores of overall uncertainty, magnitude of loss, and project risk were examined. A project's overall uncertainty score was calculated as follows. First, an overall uncertainty score was calculated by transforming each of the 23 variables to a 0-1 scale, and taking their average. The transformation was performed by dividing the score on each variable by the scale's maximum (for semantic differential scales) or the maximum value observed in the sample (for ratio scale variables). It should be noted that other approaches to calculating the overall uncertainty score could have been used. For example, factor scores associated with each variable could have been used as weights in calculating the uncertainty score of each project. Alternatively, the five uncertainty factors could have been treated as equal and factor totals could have been averaged. However, given the lack of justification for preferring one approach over the other, it appeared more parsimonious to average all uncertainty variables. A project's magnitude of loss score was found by taking the average of the eleven items forming this scale, and dividing it by the scale's maximum, 77. Finally, consistent with the risk definition adopted, the project's risk score was calculated by multiplying its uncertainty score with its magnitude of loss score.

The correlations between the criterion variable of overall risk, and overall uncertainty, magnitude of loss, and project risk were 0.43 ($p < 0.001$), 0.23 ($p < 0.01$), and 0.43 ($p < 0.001$), respectively. These results provide further evidence for criterion-related validity of the measure of software development risk proposed in this study.

USING THE RISK MEASURE

Organizations wishing to assess the risk of individual projects can use the measure developed in this study and presented in the appendix. However, as there are no established norms with which to compare project risk scores, such evaluations would initially have to be made in comparison to the sample used in this study. For that purpose, the risk score distribution of the sample's 120 projects is presented in Table 10. (Table 10 omitted).

While it is generally assumed that the set of variables constituting risk remains unchanged throughout the life cycle, their levels may vary. Thus, for example, if we were to assess the risk of a project over different stages we might observe different risk scores from one stage to another. Taking this into account, Table 10 presents percentile risk scores according to five life cycle stages. As can be seen, the mean risk score increases from

preliminary analysis to detailed analysis, but decreases during the remaining stages. However, the differences across stages are not statistically significant.

The sample project risk scores range from a minimum of 0.05 to a maximum of 0.55, with a mean of 0.26 and a standard deviation of 0.1. Assuming that the sample is representative, a project with a risk score above 0.26 would be considered risky compared with an average project. Similar to a pediatrician's use of standard charts for assessing a child's growth, the percentile risk distribution of the study sample can be used by project managers to assess and compare the degree of risk in specific projects. A project with a risk score in the 90th percentile would require greater managerial attention than a project in the 10th percentile. While an overall risk score would provide an indication of a project's inherent risk as compared with other projects, it would also be useful to study its score on each uncertainty factor and its magnitude of loss score. Such an analysis would permit project leaders to highlight the more important sources of risk for a particular project, thereby enabling them to choose appropriate project management methods and tools.

CONCLUSION

Measuring the risk of a software development project, with a view to better managing it, is an important issue. Unfortunately, not too much progress in the way of theoretical development has been made in this area. Our review of the literature indicated that this was due in part to a lack of formal definition of software development risk, and in part to the lack of reliable and valid instruments for measuring this construct.

In an attempt to respond to this need, this paper proposed a formal definition of the concept of software development risk and developed an instrument for its measurement. The variables comprising the instrument were derived from previous research in this area, and the resulting instrument has been empirically tested in a survey of 120 software development projects. The results of the study provide strong evidence of face, content, and criterion validity.

However, one important limitation of this study is the fact that it does not relate the various risk components to project outcome. The establishment of such links would be an important step forward since it is likely that different risk variables might be differentially important in influencing project outcome. A second improvement to the proposed measure would be to relate the risk variables to specific project outcomes such as budget/schedule overruns, user interface shortfalls, or poor quality software. In this way, important variables influencing different outcomes would be identified, making the proposed measure more useful.

Another important future step would be the analysis of the relationships between the proposed risk measure and other constructs related to project management. This would enable the establishment of a nomological network incorporating the notion of risk, and thus help assess the construct validity of the measure proposed. One such area in need of investigation is the relationship between project risk, project management practices, and project success. Another area that can be examined would be the study of the links between project risk and estimations of project effort. The instrument proposed here provides an important first step toward better understanding and managing software development efforts.

APPENDIX: MEASURES OF STUDY VARIABLES

Note: Items marked with an asterisk need to be reversed when calculating the variable score.

TECHNOLOGICAL NEWNESS

1 NEED FOR NEW HARDWARE (1-item binary scale, respondent: project leader)

The new system will require the acquisition and installation of new hardware.

2 NEED FOR NEW SOFTWARE (1-item binary scale, respondent: project leader)

The new system will require the acquisition and installation of new software.

3 NUMBER OF HARDWARE SUPPLIERS (1-item ratio scale, respondent: project leader)

How many hardware suppliers are involved in the development of this system?

4 NUMBER OF SOFTWARE SUPPLIERS (1-item ratio scale, respondent: project leader)

How many software suppliers are involved in the development of this system?

5 NUMBER OF USERS OUTSIDE THE ORGANIZATION (1-item ratio scale, respondent: project leader)

Approximately how many people external to the organization will be using this system (examples of external users would be customers using an automated bank teller machine, or an airline reservation system)?

APPLICATION SIZE

6 NUMBER OF PEOPLE OF TEAM (1-item ratio scale, respondent: project leader)

How many people are there on the project team?

7 RELATIVE PROJECT SIZE (3-item Much Lower Than Average/Much Higher Than Average 7-point semantic-differential scale, respondent: project leader)

7a. Compared to other information system project developed in your organization, the scheduled number of person-days for completing this project is:

7b. Compared to other information system projects developed in your organization, the scheduled number of months for completing this project is:

7c. Compared to other information system projects developed in your organization, the dollar budget allocated to this project is:

8 TEAM DIVERSITY (1-item 4-point interval scale, one point added for each category checked, respondent: project leader)

The project team members fall into which of the following groups (you can check more than one):

Information system or data processing staff

Outside consultants

Users

Others

9 NUMBER OF USERS IN THE ORGANIZATION (1-item ratio scale, respondent: project leader)

Once it is implemented, how many employees of this organization will be using this system?

10 NUMBER OF HIERARCHICAL LEVELS OCCUPIED BY USERS (1-item ratio scale, respondent: project leader)

What is the total number of different hierarchical levels occupied by the employees who will be using this system (for example, office clerks, supervisors, and managers each occupy different hierarchical levels in an organization)?

EXPERTISE

11 LACK OF DEVELOPMENT EXPERTISE IN TEAM (4-item 7-point No Expertise/Outstanding Expertise Likert scale, respondent: project leader)

Please evaluate the team's level of expertise in terms of the following:

11a. Development methodology used in this project

11b. Development support tools used in this project (e.g., DFD, flowcharts, ER model, CASE tools)

11c. Project management tools used in this project (e.g., PERT charts, Gantt diagrams, walkthroughs, project management software)

11d. Implementation tools used in this project (e.g., programming languages, data base inquiry languages, screen generators)

12 TEAM'S LACK OF EXPERTISE WITH APPLICATION (1-item 7-point semantic-differential scale, respondent: project leader)

The members of the development team are

Very familiar with this type of application/Unfamiliar with this type of application

13 TEAM'S LACK OF EXPERTISE WITH TASK (4-item 7-point No Expertise/Outstanding Expertise Likert scale, respondent: project leader)

Please evaluate the team's level of expertise in terms of the following:

13a. Overall knowledge of organizational operations

13b. In-depth knowledge of the functioning of user departments

13c. Overall administrative experience and skill

13d. Expertise in the specific application area of the system

14 TEAM'S LACK OF GENERAL EXPERTISE (6-item 7-point Low/Outstanding Likert scale, respondent: project leader)

Please evaluate the overall ability of the development team in terms of the following factors:

14a. Ability to work with undefined elements and uncertain objectives

14b. Ability to work with top management

14c. Ability to work effectively in a team

14d. Ability to successfully complete a task

14e. Ability to understand the human implications of a new information system

14f. Ability to carry out tasks quickly

15 LACK OF USER EXPERTISE AND SUPPORT (15-item 7-point Strongly Disagree/Strongly Agree Likert scale, respondent: project leader)

Generally speaking, the users of the this application:

* 15a. Have a positive opinion regarding the way in which the system can meet their needs

* 15b. Feel they need computerized support in carrying out the tasks for which the system is being developed

15c. Are not enthusiastic about the project

15d. Have negative attitudes regarding the use of computers in their work

* 15e. Are ready to accept the various changes the system will entail

15f. Do not actively participate in requirement definition

* 15g. Are available to answer the development team's questions

* 15h. Are aware of the importance of their role in successfully completing the project

15i. Are not very familiar with information system development tasks and life cycle stages

* 15j. Are an integral part of the development team

15k. Are not very familiar with data processing as a working tool

15l. Have little experience with the activities to be supported by the future application

* 15m. Quickly respond to development team requests (for information, comments, approvals)

* 15n. Will have no constraints in fulfilling their development responsibilities with respect to this system

15o. Are not very familiar with this type of application

APPLICATION COMPLEXITY

16 TECHNICAL COMPLEXITY (3-item Slightly Complex/Highly Complex 7-point semantic-differential scale, respondent: project leader)

Referring to the application being developed, how would you evaluate the technical complexity of each of the following elements:

16a. The hardware (computers, networks)

16b. The software

16c. The database

17 NUMBER OF LINKS TO EXISTING SYSTEMS (1-item ratio scale, respondent: project leader)

How many existing information systems will be linked to this system?

18 NUMBER OF LINKS TO FUTURE SYSTEMS (1-item ratio scale, respondent: project leader)

How many information systems currently under development will be linked to this system?

ORGANIZATIONAL ENVIRONMENT

19 EXTENT OF CHANGES BROUGHT (4-item 7-point semantic-differential scale, respondent for 3a and 3b: project leader, for 3c and 3d: user representative)

19a. The development of this system will require that user tasks be modified:

Only slightly/A great deal

19b. In general, this application will lead to:

Few changes in the organization/Major changes in the organization

19c. The development of this system will require that user tasks be modified:

Only slightly/A great deal

19d. In general, this application will lead to:

Few changes in the organization/Major changes in the organization

20 RESOURCE INSUFFICIENCY (3-item More Than Enough/Extremely Insufficient 7-point semantic-differential scale, respondent: project leader)

20a. In order to develop and implement this system, the scheduled number of person-days is:

20b. In order to develop and implement this system, the scheduled number of months is:

20c. In order to develop and implement this system, the dollar budget provided is:

21 INTENSITY OF CONFLICTS (6-item 7-point semantic-differential scale, respondent for 21a, 21b, and 21c: project leader, for 21d, 21e, and 21f: user representative)

With the framework of this project: conflicts between team members:

21a. Rarely occur/Frequently occur

21b. Are not very serious/Are very serious

21c. Concern relatively unimportant matters/Concern very important matters

Within the framework of this project, conflicts between the users and the team members:

21d. Rarely occur/Frequently occur

21e. Are not very serious/Are very serious

21f. Concern relatively unimportant matters/Concern very important matters

22 LACK OF CLARITY OF ROLE DEFINITIONS (3-item 7-point semantic-differential scale, respondent: project leader)

22a. The role of each member of the project team is:

Clearly Defined/Not Clearly Defined

22b. Communications between those involved in the project are:

Pleasant/Unpleasant

22c. The role of each person involved in the project is:

Clearly Defined/Not Clearly Defined

23 TASK COMPLEXITY (20-item 7-point semantic-differential scale, respondent: user representative)

23a. The sequence of steps of be carried out to successfully complete these activities is:

Easy to identify/Hard to identify

23b. While the consequences of some activities are easy to predict, others are often unpredictable. The consequences of the activities in question are:

Easy to predict/Hard to predict

23c. A well-defined body of knowledge on which to base the execution of these activities:

Exists/Does not exist

24d. In general, one can determine whether or not the activities were successfully performed:

Immediately/After a long period of time

23e. When problems arise in carrying out these activities, getting help is:

Easy/Difficult

23f. When carrying out these activities, problems which cannot be immediately resolved arise:

Rarely/Frequently

23g. Solving these problems typically requires:

Little time/A lot of time

23h. In your opinion, these activities are:

Routine/Always new

* 23i. In general, carrying out these activities requires the use of:

A large number of methods and procedures/A small number of methods and procedures

23j. These rules and procedures are:

Rarely subject to change/Frequently subject to change

* 23k. Carrying out these activities requires:

A large number of different steps/A small number of different steps

* 23l. These activities can be performed in:

Many different ways/Only one way

23m. Carrying out these activities generally involves:

A large number of repetitive tasks/A small number of repetitive tasks

23n. When carrying out these activities, the extent of variety with respect to situations, actors, and tasks is:

Low/High

23o. Regardless of the actors or the specific situations, the tasks and the procedures involved in carrying out these activities are:

Always the same/Extremely varied

23p. In carrying out these activities:

There is a single objective to reach/There are multiple objectives to reach

23q. When carrying out these activities all objectives:

Can be reached/Cannot be reached

23r. When choosing a specific way to proceed:

One knows what the result will be/One does not know what the result will be

23s. When evaluating the way in which all of these activities were carried out, the measure of their success is based on:

One criterion/Several criteria

*23t. Carrying out these activities depends on the execution of:

Many other related activities/Only a few other related activities

24 MAGNITUDE OF POTENTIAL LOSS (11-item 7-point Little Impact/Large Impact Likert scale, respondent: project leader)

If, for some reason, the information system being developed is not implemented or if it has operational problems, what impact would this have on your organization in terms of the following:

24a. Customer relations

24b. Financial health

24c. Reputation of the information system department

24d. Profitability

24e. Competitive position

24f. Organizational efficiency

24g. Organizational image

24h. The survival of the organization

24i. Market share

24j. Reputation of the user department

24k. Ability to carry out current operations

REFERENCES

1. Alloway, R.M. Temporary management systems: application of a contingency model to the creation of computer based information systems. Ph.D. dissertation, Sloan School of Management, MIT, 1976.
2. Alter, S. Implementation risk analysis. TIMS Studies in Management Sciences, 13, 2 (April 1979), 103-119.
3. Alter, S., and Ginzberg, M. Managing uncertainty in MIS implementation. Sloan Management Review, 20, 1 (Fall 1978), 23-31.
4. Altman, E.I. Financial ratios, discriminant analysis and the prediction of corporate bankruptcy. Journal of Finance, 23, 4 (September 1968), 589-609.
5. Anderson, J., and Narasimhan, R. Assessing implementation risk: a methodological approach. Management

Science, 25, 6 (June 1979), 512-521.

6. Bariff, M.L. Information requirements analysis: a methodological review. Working paper 76-08-82, Wharton School, University of Pennsylvania, 1977.
7. Barki, H. A contingency model of DSS success: an empirical investigation. Ph.D. dissertation, School of Business Administration, The University of Western Ontario, London, Ontario, 1983.
8. Beath, C.M. Strategies for managing MIS: a transaction cost approach. Proceedings of the Fourth International Conference on Information Systems, Houston, ACM/SIGBDP, December 1983, pp. 415-427.
9. Bell, T.E. Managing Murphy's law: engineering a minimum-risk system. IEEE Spectrum, 26, 6 (June 1989), 24-27.
10. Berkeley, D.; Humphreys, P.C.; and Thomas, R.D. Project risk action management. Construction Management and Economics, 9 (1991), 3-17.
11. Bernier, C. Etude des relations entre l'environnement, l'incertitude environnementale perçue et le mode de gestion dans les projets de développement de systèmes d'information. Ph.D. dissertation, Ecole des Hautes Etudes Commerciales, Montreal, Quebec, 1989.
12. Bernier, C., and Rivard, S. A contingency model for the management of software projects: an empirical investigation. Proceedings of the ASAC Conference, Whistler, British Columbia, 11, 4 (June 1990), 23-38.
13. Boehm, B.W. Software Risk Management. Los Alamitos, CA: IEEE Computer Society Press, 1989.
14. Brecher, A. An overview of formal methods of risk assessment. Conference Record IEEE Electro, Boston, 1988, pp. 1-9.
15. Casher, J.D. How to control risk and effectively reduce the chance of failure. Management Review, 73, 6 (June 1984), 50-54.
16. Charette, R.N. Software Engineering Risk Analysis and Management. New York: McGraw-Hill, 1989.
17. Davis, G.B. Strategies for information requirements determination. IBM System Journal, 21, 1 (March 1982), 4-30.
18. Denenberg, H.S.; Eilers, R.D.; Melone, J.J.; and Zelten, R.A. Risk and Insurance, 2d ed. Englewood Cliffs, NJ: Prentice-Hall, 1974.
19. Galbraith, J. Organization Design. Reading, MA: Addison-Wesley, 1977.
20. Guarro, S.B. Principles and procedures of the LRAM approach to information systems risk analysis and management. Computers & Security, 6, 6 (December 1987), 493-504.
21. Haimes, Y.Y. Total risk management. Risk Analysis, 11, 2 (1991), 169-171.
22. Kangari, R., and Boyer, L.T. Risk management by expert systems. Project Management Journal, 20, 1 (1989) 40-48.
23. Kaplan, S., and Garrick, J.B. On the quantitative definition of risk. Risk Analysis, 1, 1 (1981), 11-27.
24. Kim, J., and Mueller, C.W. Factor Analysis. Statistical Method and Practical Issues. Beverly Hills, CA: Sage Publications, 1988.
25. Linnerooth-Bayer, J., and Wahlstrom, B. Applications of probabilistic risk assessments: the selection of appropriate tools. Risk Analysis, 11, 2 (1991), 239-248.
26. McFarlan, F.W. Portfolio approach to information systems. Harvard Business Review, 59, 5 (September-October 1981), 142-150.
27. Naumann, J.D.; Jenkins, A.M.; and Wetherbe, J.C. Empirical investigation of systems development practices and results. Working paper 83-09, University of Minnesota, Minnesota, 1983.
28. Post, G.V., and Diltz, D.J. A stochastic dominance approach to risk analysis of computer systems. MIS Quarterly, 10, 4 (December 1986), 363-375.
29. Rainer, R.K. Jr.; Snyder, C.A.; and Can; H.H. Risk analysis for information technology. Journal of Management Information Systems, 8, 1 (1991), 129-147.
30. Rivard, S.; Talbot, J.; and Charest, M. Measuring the uncertainty of a system development project. Proceedings of the ASAC Conference, Toronto, Ontario, 8, 4 (June 1987), 115-124.
31. Robey, D.; Farrow, D.L.; and Franz, C.R. Group process and conflict in system development. Management

Science, 35, 10 (1989), 1172-1191.

32. Rothfeder, J. It's late, costly, incompetent—but try firing a computer system. Business Week (November 7, 1988), 164-165.

33. Sherer, S.A. Measuring the risk of software failure: a financial application. Proceedings of the Tenth International Conference on Information Systems, Boston, ACM/SIM/TIMS, December 1989, 237-245.

34. Tversky, A., and Kahneman, D. Belief in the law of small numbers. In D. Kahneman, P. Slovic, and A. Tversky, A. (eds.), *Judgement under Uncertainty: Heuristics and Biases*. Cambridge: Cambridge University Press, 1982, pp. 23-31.

35. U.S. Air Force Command. Software risk abatement. AFSC/AFLC Pamphlet 8045, Andrew's Air Force Base, CA: September 30, 1988.

36. Van de Ven, A.H., and Delbecq, A. A task contingent model of work-unit analysis. *Administrative Science Quarterly*, 19, 2 (June 1974), 183-197.

37. Van de Ven, A.H., and Ferry, D.L. *Measuring and Assessing Organizations*. New York: Wiley Interscience, 1980.

38. Wideman, R.M. Risk management. *Project Management Journal*, 17, 4 (September 1986), 20-26.

39. Zmud, R.W. Management of large software development efforts. *MIS Quarterly*, 4, 2 (June 1980), 45-55.

Acknowledgments: This study was funded by CEFRIO (Centre francophone de recherche en informatisation des organisations) and the Social Sciences and Humanities Research Council of Canada. The authors would like to thank the reviewers for providing valuable comments that contributed to the improvement of the quality of this article.

Henri Barki is Associate Professor of Information Systems at the Ecole des Hautes Etudes Commerciales in Montreal. He received his Ph.D. in information systems from the School of Business Administration, University of Western Ontario. His research interests, including the study of user participation, user involvement, and conflicts, center on the management of software development projects. His papers have been published in *MIS Quarterly*, *Information and Management*, *Canadian Journal of Administrative Sciences*, and *INFOR*.

Suzanne Rivard is a Professor of Information Systems at the Ecole des Hautes Etudes Commerciales in Montreal. She holds an M.B.A. from Ecole des Hautes Etudes Commerciales and a Ph.D. in MIS from the University of Western Ontario. Her main research interests include the management of software development projects, end-user computing, and impacts of information technologies. She has published articles in a variety of journals, including *Communications of the ACM*, *MIS Quarterly*, *Information and Management*, *Interfaces*, and *INFOR*.

Jean Talbot is Assistant Professor of Information Systems at the Ecole des Hautes Etudes Commerciales in Montreal. He received his doctorate in information systems from the Universite Montpellier II in France. His current research interests are in the area of system development management.

DETALHES

Assunto:	Systems development; Systems design; Risk assessment; Methods
Classificação:	5240: Software & systems
Título:	Toward an assessment of software development risk
Autor:	Barki, Henri; Rivard, Suzanne; Talbot, Jean
Título da publicação:	Journal of Management Information Systems: JMIS; Armonk
Volume:	10

Edição:	2
Páginas:	203
Número de páginas:	23
Ano de publicação:	1993
Data de publicação:	Fall 1993
Editora:	M. E. Sharpe Inc.
Local de publicação:	Armonk
País de publicação:	United States, Armonk
Assunto da publicação:	Political Science
ISSN:	07421 222
Tipo de fonte:	Scholarly Journals
Idioma de publicação:	English
Tipo de documento:	PERIODICAL
Número de registro:	00813256
ID do documento ProQuest:	218964505
URL do documento:	https://search.proquest.com/docview/218964505?accountid=26645
Copyright:	Copyright M. E. Sharpe Inc. Fall 1993
Última atualização em:	2016-02-20
Base de dados:	Technology Collection

LINKS:

Copyright da base de dados © 2019 ProQuest LLC. Todos os direitos reservados.

[Termos e condições](#) [Entre em contato com o ProQuest](#)