The effects of development process modeling and task uncertainty on development quality performance

Arun Rai a,*, Hindi Al-Hindi b

a Electronic Commerce Institute, Robinson College of Business, Georgia State University, Atlanta, GA 30303, USA
b Department of Quantitative methods, College of Business and Economics, King Saud University-Al Quasseem Branch, Al Melaida P.O. Box 6033, Saudi Arabia

Received 24 March 1999; accepted 19 September 1999

Abstract

We examine the impact of development process modeling on outcomes in software development projects, limiting our attention to process and product quality. Modeling the software development process requires a careful determination of tasks and their logical relationships. Essentially, the modeling is undertaken to establish a management framework of the project. We define and interrelate development process modeling, task uncertainty, and development outcomes, as assessed by product and process quality. A survey-based research design was used to collect data to prove our model. The results suggest that development process modeling is positively related to both product and process quality, while task uncertainty is negatively related to them. Development process modeling reduces the negative impact of task uncertainty on quality-oriented development outcomes. Development projects operating with high levels of task uncertainty should consider defining development process models that provide a framework for management of the project by establishing tasks and their logical interrelationships. Such a model should promote shared understanding of the work process among development constituents and enhance resource utilization efficiency. © 2000 Elsevier Science B.V. All rights reserved.

Keywords: Management information systems; Systems development methodologies; Process quality; Product quality

1. Introduction

The demand for many new and different information system (IS) applications has increased the scope and complexity of the development process. Despite advances in the tools available for development, productivity is still low [4], development costs remain high, and it continues to be a challenge to develop quality products [42,46]. Even after significant resources are poured into software development projects, many end up failing [23].

While poor performance and outright failures can be attributed to a number of factors, the management approach applied to development projects is seen as being a major cause of these pitfalls. Researchers in the area recognize that carefully conceived management practices should reduce development costs and improve development outcomes [26,50]. Technology-focused solutions have been faulted for these problems, while limited attention has been paid to examining how these problems can be alleviated by rethinking management approaches being used [1]. Recent work by Ravichandran and Rai [41]
emphasizes the importance of developing an organizational systems perspective for the management of systems development [41].

The domain of our investigation is software development projects. Their management approach is usually based on the experience of the project manager [16]. In fact, many software projects are carried out in an ad hoc fashion without a well-established management framework. Old methods, such as the waterfall model and rapid prototyping, were proposed as frameworks to guide the management of the development process. However, each project is embedded in a specific context and has its unique set of information requirements, technological issues, resource constraints, and other contextual factors. While generic frameworks are useful, they have to be modified, adapted, and then instantiated in their context. They may be inadequate if organizations do not apply them to create process models that could enforce discipline within tasks, establish standardized interfaces between tasks, and improve the predictability associated with resource requirements.

Projects differ significantly. The requirements for some may be fuzzy, for others volatile, and for yet others predictable and stable. The management literature suggests that the selection and use of management practices should be informed by key contingencies underlying a given domain [29]. This requires specification of constructs defining alternate management practices, key task characteristics, and measures of performance outcome. A second step is to develop the interrelationships between these identified constructs.

The following question then seems reasonable to pose: what important contingencies should be considered when studying the impact of development process modeling? Information processing theory suggests that an organization should be designed to manage uncertainty so as to achieve acceptable levels of predictability for the work system. A number of factors can affect the tasks and some of these factors are difficult to predict with a reasonable amount of certainty [17]. For example, user–developer relationships, user participation, business domain characteristics, technology and operating system platforms on which the software applications will operate can all impact the degree of task uncertainty.

In summary, our research is directed to increase our understanding of the interrelationships between development process modeling, task uncertainty in development projects, and quality-oriented development outcomes. The specific research question is:

What are the relationships between development process modeling, task uncertainty, and quality-oriented development outcomes?

2. Research model and hypotheses

2.1. Research model

There are behavioral, economical and attitudinal outcomes associated with development projects. For reasons of scope, we limit our attention to quality outcomes. Cooprider and Henderson [12] suggest that examining product and process outcomes together can reveal differential impacts of them on quality. Past research has identified some attributes of development quality, such as effective coordination [35], project completion within schedule and budget [37], and overall quality of the development efforts [2]. We define development process quality as the degree to which the process is designed to promote consensus among people participating in the development process, operate within established resource parameters, and reduce waste and redundancy. We define software product quality as the overall evaluation of the final product produced by the development process. We consider objective criteria, such as reliability and maintainability of the product, and subjective criteria, such as user acceptance and satisfaction, as part of the overall assessment.

The management process should have a direct impact on development quality, but it can also indirectly impact these outcomes by maximizing benefits obtained from development tools used and personnel participating in the process [27]. There is little empirical evidence of the impact of process modeling on development quality. We examine whether process and product quality outcomes vary by the level of process modeling undertaken in a development project. Increased levels of process modeling should reduce task uncertainty while increases in task uncertainty should negatively impact process and product quality. The examination of these direct relationships is complemented by an examination of the nature of the interaction between software process...
modeling and task uncertainty. Our research model is schematically represented in Fig. 1.

2.2. Development process modeling and quality-oriented development outcomes

A software development process is a collection of interrelated activities performed in a logical order to produce an application system. A common understanding of the software development process among participants, including developers, managers, and users, can be very useful for planning and controlling development activities [9,13,32]. However, many projects are carried out without adequate planning, with a poor explication of the overall development process, and the absence of a well-conceived managerial framework. Inadequate planning is one of the reasons for the high failure rates [6,7]. The task interfaces are ill-defined, deadlines are vague and unrealistic, resources required have not been rationalized, and scheduling and resource allocation models are not poorly developed.

Development process modeling can be an important step in building the management infrastructure necessary to manage the process [28,34]. Detailed development tasks determined during process modeling represent milestones that can be used for planning, scheduling and control.

Development and implementation of a process model is increasingly important as development units need to shift their focus from the traditional view of focusing on quality of the final product to a more comprehensive view of improvement by focusing on both product and process quality [18]. Continuous improvement of the development process is considered essential for producing high-quality applications [36]. Process modeling can be used to analyze, understand and improve the design of the current development process. The concept of ‘quality at the source’ suggests that deliverables at each stage should be examined and inspected before being transferred to subsequent stages. This leads to the following hypotheses:

\[ H_1: \text{The degree to which a process model has been established for a development project is positively related to process quality.} \]

\[ H_2: \text{The degree to which a process model has been established for a development project is positively related to product quality.} \]

2.3. Development process modeling and task uncertainty

Process models can lead to a number of benefits, including disciplining the process and reducing uncertainty [20]. Modeling detailed development tasks in a formal fashion is an effective way to reduce role ambiguity among participants. Lack of this model is likely to have negative consequences on development outcomes [30]. Process models can also reduce uncertainty by clearly defining details of development tasks; then the association of tasks with process variations can focus the attention of development units on early identification of problems.

Individuals performing a particular development task interpret task requirements according to their own views [51]. Furthermore, lack of communication among participants is a major cause of failure [14]. Formal and impersonal coordination, such as plans, schedules and documents allow effective coordination with positive impact on software development in terms of client satisfaction, quality, and productivity. As noted by Charette [10], a well-defined development process can help software developers in their interpretation of their tasks, and should reduce uncertainty in software development [5]. This leads to the hypothesis:

\[ H_3: \text{The degree to which a process model has been established for a development project is negatively related to the degree of task uncertainty.} \]

2.4. Task uncertainty and quality-oriented development outcomes

Generally speaking, high levels of task uncertainty are associated with software development [33]...
exacerbated by hard-to-predict factors impacting different stages in the development process. Ongoing changes in user requirements contribute to the high levels of uncertainty associated [11,26]. Application development teams often include people with limited knowledge about the problem domain and detailed knowledge is often provided too late to help [39]. Inadequate information can result in decisions to delay certain steps or to execute the development process in a trial and error fashion [15]. This leads to the hypotheses:

**H4**: The degree of task uncertainty in a development project is negatively related to development process quality.

**H5**: The degree of task uncertainty in a development project is negatively related to development product quality.

2.5. Development process modeling-task uncertainty interaction

The interaction relationships were proposed to show how development process modeling impacts the relationship between task uncertainty and development quality: the assumption being that organization performance is dependent on the organization’s ability to handle uncertainty through its information processing capability [43,47]. To achieve a maximum level of performance, compromise must occur between an organization’s information processing capability and the level of uncertainty that it faces.

Several mechanisms have been suggested to increase the information processing capability of the development subunit. These include coordination [38], standardization, rethinking the decision making structure, and modern development practices. Accordingly, development process modeling should be more important in projects characterized by higher levels of task uncertainty. This leads to the hypotheses:

**H6**: The interaction between modeling and task uncertainty influences development process quality.

**H7**: The interaction between modeling and task uncertainty influences development product quality.

3. The empirical study

3.1. Data collection

To obtain data for this study, a survey instrument was designed, pre-tested and sent to 1050 senior IS executives located across the United States. The names of these executives and their companies were randomly selected from the Directory of Top Computer Executives. This random selection process was used as we wanted to test our hypotheses using data about projects emanating from diverse organizational and industrial contexts. The questionnaire clearly stated that our target was a recently completed software development project within the firm (Appendix A). An accompanying letter explained this and requested the senior IS executive to forward all materials to the project manager or a well-informed member of the project team who participated in the development process. This is consistent with Huber and Powell’s [25] recommendation that the person(s) most knowledgeable should be chosen as respondents. A reminder letter along with a follow-up questionnaire was sent 4 weeks after the initial mailing. A total of 131 responses were received; 36 organizations indicated that they did not develop software in-house. The remaining 95 completed the questionnaire, representing a response rate of 12.4%. This is normal for such surveys.

Given the low response rate, nonresponse bias, if present, can be a potential threat to the external validity of the study. External validity is compromised when respondents differ significantly from nonrespondents on important characteristics [19]. We conducted a series of commonly advocated tests to examine for nonresponse bias. Firm size and environmental characteristics, such as industry membership, have been shown to affect the result [8,40]. No differences were detected in size assessed by number of employees, and industry composition between respondents and nonrespondents. This suggested a lack of nonresponse bias. Differences in mean values of key variables were compared across waves of respondents: responses received later can be considered relatively similar to nonrespondents. No significant differences were detected between early and late respondents further alleviating our concerns. The lack of significant differences suggests that
responses can be pooled without any loss of generalizability.

We also tested to see if our sample was biased with respect to key characteristics, such as size of development team, project duration, and project budget. Overall, the 95 projects showed a good dispersion of project context characteristics. An examination of the distribution of the dependent variables, product quality and process quality, did not reveal any biases as their means and medians were similar, skewness was less than two, and kurtosis was less than five [21]. Thus, while our response rate is low, the series of tests did not reveal any significant threats to the generalizability of findings to the population of inhouse developed software development projects. A profile of the projects is summarized in Table 1, while Table 2 provides a summary of the development tools used in these development projects.

### 3.2. Instrument development

The questionnaire for this study, which is part of a larger research effort on software development management practices, was developed through an extensive review of the literature. The first section of the questionnaire asked respondents about general characteristics of their IS organization and the specific development project. The second solicited information about their management practices and the nature of the development task. A 7-point Likert-type scale that ranged from strongly disagree to strongly agree was used for each of the items. The third section listed items to evaluate development outcomes. The respondents were asked to indicate their project performance on a 7-point Likert type scale. The questionnaire was critiqued by three researchers who have worked extensively in the area of IS development and by two individuals with significant practical experience in the management of development projects. Their comments were addressed prior to our data collection.

#### 3.3. Measures for independent variables

##### 3.3.1. Development process modeling

To the best of our knowledge, an operational measure has not been developed to assess the degree to which process models have been established for a development project. We recognize that several techniques, including Petri nets, state transition, and quantitative models can be used to model processes [13]. Regardless of the technique used, process models should be effective tools. Our focus here is not on the specific modeling technique used. We developed a 6-item scale to assess the degree to which a process model has been established. These items include definition of key tasks: level of detail of tasks, relationships between development tasks, use of a process, and the extent to which the organization enforces the use of a software process model.

Stewart [45] suggested that Bartlett’s test of sphericity and the Kaiser-Meyer-Olin measure of sampling adequacy be examined to assess whether or not a set of variables are appropriate for factor analysis. The test assesses whether the correlation matrix comes from a

---

**Table 1**

<table>
<thead>
<tr>
<th>Characteristics</th>
<th>Mean</th>
<th>Standard Deviation</th>
<th>Minimum</th>
<th>Maximum</th>
</tr>
</thead>
<tbody>
<tr>
<td>Size of development team</td>
<td>8.24</td>
<td>11.17</td>
<td>1</td>
<td>75</td>
</tr>
<tr>
<td>Number of distinct user department involved</td>
<td>3.45</td>
<td>2.35</td>
<td>1</td>
<td>11</td>
</tr>
<tr>
<td>Project duration (months)</td>
<td>10.7</td>
<td>8.65</td>
<td>1</td>
<td>36</td>
</tr>
<tr>
<td>Project budget (thousands $)</td>
<td>571.77</td>
<td>907.8</td>
<td>4</td>
<td>5000</td>
</tr>
<tr>
<td>Number of distinct computer programs developed</td>
<td>112.7</td>
<td>208</td>
<td>2</td>
<td>1000</td>
</tr>
<tr>
<td>Number of full-time employees in the IS department</td>
<td>34.3</td>
<td>61.2</td>
<td>1</td>
<td>350</td>
</tr>
</tbody>
</table>

**Table 2**

<table>
<thead>
<tr>
<th>Tools</th>
<th>Percentage of projects</th>
</tr>
</thead>
<tbody>
<tr>
<td>CASE Tools</td>
<td>15.7</td>
</tr>
<tr>
<td>Data base management systems</td>
<td>67.4</td>
</tr>
<tr>
<td>Fourth generation languages</td>
<td>34.8</td>
</tr>
<tr>
<td>Other tools</td>
<td>46.1</td>
</tr>
</tbody>
</table>

\[a\] Some firms use multiple tools.
population of variables that are independent. Measure of sampling adequacy (MSA) provides a measure of the extent to which variables belong together. Kaiser and Rice [31] provided a calibration of the MSA measure with regard to the degree of appropriateness of using factor analysis on a set of measurement items. They classified a value of 0.80 as ‘meritorious’, values between 0.6 and 0.8 as ‘mediocre’, and values below 0.50 as ‘unacceptable.’

For the measurement items being considered for the development process modeling construct, Bartlett’s test of sphericity led to a rejection of the null hypothesis of variable independence at a level of significance of 0.000. The measure of sampling adequacy was computed to be 0.83 and suggested that the items were appropriate for factor analysis. Principal component analysis followed by a varimax rotation resulted in a two-factor solution. Five items used to define the development process modeling scale loaded on the first factor, each with loadings greater than 0.7 (Table 3). These items reflect the degree of process modeling performed for the development project. The second factor, with one item loading significantly on it, suggests that awareness of the organization about the importance of process modeling is a separate construct. Thus, we limit our measure of development process modeling to the five items that loaded on the first factor. There were no sudden drops in the item-total correlation, and a Cronbach’s alpha value of 0.89 suggests a high level of internal consistency among measurement items.

3.3.2. Task uncertainty

A 3-item instrument was used to measure uncertainty. These items reflect fluctuation in users’ requirements, information available to perform the tasks, and the extent to which the project was subject to uncertain events. These items were adapted from Goodhue and Thompson [22] and Van de Ven, et al. [48]. As expected, Bartlett’s test of sphericity led to a rejection of the null hypothesis of variable independence at a level of significance of 0.000. The measure of sampling adequacy was computed to be 0.60. Not surprisingly, the principal component analysis resulted in all three items loading on one factor, with two of the items having loadings greater than 0.80 and one item having a loading of 0.63 (Table 4). There were no sudden drops in the item-total correlation and a Cronbach’s alpha value of 0.63 suggests a reasonable level of internal consistency among measurement items.

3.4. Dependent variables: measures for development quality

3.4.1. Product quality

A 5-item scale was used to measure software product quality: reliability, flexibility, maintainability,
system acceptance, and satisfaction. The instrument used was adapted from Alavi [2], Henderson and Lee [24], and Shin and Lee [44]. As expected, the null hypothesis of variable independence of these was rejected at a level of significance of 0.000. Our MSA measure was 0.83 and suggested that the five items were appropriate for factor analysis. Principal component analysis resulted in a one-factor solution (see Table 5). Factor analysis results for the development product quality construct showed that the five items loaded on one factor, each with loadings greater than 0.70. The item-total correlations indicated no sudden drops providing evidence of homogeneity among the items and a Cronbach’s alpha value of 0.89 reveals a high level of internal consistency among measurement items.

3.4.2. Process quality

A 5-item scale was used to assess the quality of the development process. These items reflect some desirable characteristics of the process such as the degree to which the project was completed on schedule, the degree to which the project met cost targets, and the degree of agreement among participants. The items used were adapted from Alavi [2], Henderson and Lee [24] and Mahmood [37]. Batlett’s test of sphericity led to a rejection of the null hypothesis of variable independence at a level of significance of 0.000. The measure of sampling adequacy was computed to be 0.74 and suggested that the items were appropriate for factor analysis. Principal component analysis resulted in a one-factor solution (Table 6). However, one item has a loading of 0.37, and was deleted. The remaining items were used to determine a measure for process quality. There were no sudden drops in the item-total correlation, and a Cronbach’s alpha value of 0.78 suggests a high level of internal consistency among the four measurement items.

3.5. Convergent and discriminant validity

Convergent validity is achieved if items that measure the same construct are highly correlated. One-tailed t-test was used to assess the significance of inter-item correlations: the test revealed that are significant at a level of significance of 1%. Examination of the correlation between an item and each of the constructs was used to assess discriminant validity. Items of a particular construct have discriminant validity when they have higher correlations with their construct than with other constructs. The item-construct correlations showed that all measurement items have higher correlations with their construct than with others. Based on these results, our constructs and their measures have acceptable levels of convergent and discriminant validity.

3.6. Summary of psychometric properties

Venkatraman and Grant [49] recommended that survey instruments used for empirical research should use scales (1) with multiple higher level items rather than single, nominal items, (2) that are internally consistent, and (3) that are valid. The measures for task uncertainty, product and process quality were adapted from previous IS research on the measurement of these constructs. The process modeling construct was developed for this study and subjected to pretest. Factor analysis of the measurement items resulted in factor structures providing evidence of
measurement validity. Three of the constructs have reliabilities greater than 0.80 and one of the constructs, task uncertainty, has a reliability of 0.63. These values are acceptable for exploratory theory testing research in the behavioral and social sciences. The scale reliabilities along with some descriptive statistics of each of the constructs are given in the Table 7. Correlation analysis between constructs and between items and constructs suggest adequate levels of convergent and discriminant validity of the measures.

4. Results and discussion

4.1. Direct relationships

The correlation coefficient between software process modeling and process quality was positive and significant ($p<0.005$), which gives support for hypothesis H1. The correlation coefficient between software process modeling and development product quality is positive and significant ($p<0.0001$), which gives support to hypothesis H2 and significant ($p<0.0001$), which gives support to hypothesis H3. The correlation coefficient between development task uncertainty and process quality was negative and significant ($p<0.005$), which supports hypothesis H4. Also, the correlation coefficient between task uncertainty and process quality was negative and significant ($p<0.005$), which supports hypothesis H5.

4.2. Interaction relationships

The interaction relationships proposed pertain to the impact of the interaction between software process modeling and task uncertainty on quality-oriented development outcomes. The first interaction pertains to the impact of process modeling on the relationship between task uncertainty and product quality; the second pertains to the effect of process modeling on the relationship between task uncertainty and development process quality. When dealing with an interaction relationship, the selection of the functional form that represents the relationship is of a significant importance. In case the outcome variable is better

Table 7
Correlation coefficients, reliabilities and descriptive statistics

<table>
<thead>
<tr>
<th>Variable</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>Cronbach's alpha</th>
<th>Mean</th>
<th>Standard deviation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Product quality</td>
<td>1.0</td>
<td></td>
<td></td>
<td></td>
<td>0.89</td>
<td>5.57</td>
<td>0.96</td>
</tr>
<tr>
<td>Process quality</td>
<td>0.77</td>
<td>1.0</td>
<td></td>
<td></td>
<td>0.78</td>
<td>4.65</td>
<td>1.18</td>
</tr>
<tr>
<td>Development process modeling</td>
<td>0.43</td>
<td>0.37</td>
<td>1.0</td>
<td></td>
<td>0.89</td>
<td>5.04</td>
<td>1.2</td>
</tr>
<tr>
<td>Task uncertainty</td>
<td>-0.35</td>
<td>-0.30</td>
<td>-0.54</td>
<td>1.0</td>
<td>0.63</td>
<td>4.86</td>
<td>1.67</td>
</tr>
</tbody>
</table>

Table 8
Regression analysis: process modeling and task uncertainty on process quality

<table>
<thead>
<tr>
<th>Independent variable</th>
<th>Interaction modela</th>
<th>Main effect modelb</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>b</td>
<td>t</td>
</tr>
<tr>
<td>Process modeling ($s$)</td>
<td>1.45</td>
<td>3.15</td>
</tr>
<tr>
<td>Task uncertainty ($t$)</td>
<td>0.98</td>
<td>2.11</td>
</tr>
<tr>
<td>Process modeling$\times$ task uncertainty ($s\times t$)</td>
<td>-0.22</td>
<td>-2.62</td>
</tr>
<tr>
<td>Model $R^2$</td>
<td>0.22</td>
<td></td>
</tr>
<tr>
<td>Adjusted $R^2$</td>
<td>0.19</td>
<td></td>
</tr>
<tr>
<td>$F$-value</td>
<td>8.38</td>
<td></td>
</tr>
<tr>
<td>$P&lt;$</td>
<td>0.0001</td>
<td></td>
</tr>
</tbody>
</table>

a $y_1=b_0+b_1 s+b_2 t+b_3 s\times t$, where $y_1$ is development process quality.

b $y_1=b_0+b_1 s+b_2 t$. 

explained by the presence of the two interacting variables than by only one, as suggested by hypotheses H6 and H7, the multiplicative form of interaction may be more appropriate [3].

To test the interaction relationship of H6, an equation with a multiplicative interaction term was estimated. Table 8 shows the results of the regression analysis for the interaction model and the main effect model for these variables: the interaction term was negative and significant. Also, the improvement of goodness of fit resulting from the interaction term (incremental $r^2$) is also significant ($\Delta r^2 = 0.06$, $F$-value $= 7.0$, $p < 0.01$). These results support hypothesis H6. The negative sign of the interaction term means that task uncertainty has a lesser negative impact on process quality at higher levels of process modeling.

Similarly, a multiplicative interaction equation was estimated to test the impact of the interaction between process modeling and task uncertainty on development product quality (H7). The results of the regression analysis (Table 9) show that the interaction term is negative and significant. Also, the incremental $r^2$ resulting from including the interaction term in the regression equation is significant ($\Delta r^2 = 0.07$, $F$-value $= 8.73$, $p < 0.01$). These results support H7.

Task uncertainty and development process modeling exhibit an interaction effect on development quality supporting the contingency relationship between the two variables as suggested by our theoretical model. Therefore, higher levels of process modeling not only improve development quality directly but also reduce the negative impacts of task uncertainty in development projects.

5. Conclusions

We used a contingency framework based on information processing theory to formulate our theoretical model. Findings support seven research hypotheses that were formulated.

Our results suggest that process modeling has a positive association with development outcome quality. Also, it has an indirect impact by reducing the negative impact of development task uncertainty on development quality; process modeling can apparently be used to define the information processing structure and capability of the project so to better manage uncertainty. Consequently, the negative impact of development task uncertainty is likely to be offset by process modeling.

It is useful to model the development process before starting development efforts: this can provide a useful managerial framework.

The correlation coefficient between enforcing the use of process models and the level of process modeling was positive and significant. This suggests that the organization represented by its top management can positively impact development process modeling by emphasizing the conceptualization and use of process models. Project managers and development personnel could conceivably be empowered to develop process models for their respective projects.

---

Table 9
Regression analysis: process modeling and task uncertainty on product quality

<table>
<thead>
<tr>
<th>Independent variable</th>
<th>Interaction model$^a$</th>
<th>Main effect model$^b$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>b</td>
<td>t</td>
</tr>
<tr>
<td>Process modeling ($s$)</td>
<td>1.24</td>
<td>3.44</td>
</tr>
<tr>
<td>Task uncertainty ($t$)</td>
<td>0.81</td>
<td>2.22</td>
</tr>
<tr>
<td>Process modeling×task uncertainty ($s×t$)</td>
<td>-0.18</td>
<td>-2.78</td>
</tr>
<tr>
<td>Model $R^2$</td>
<td>0.27</td>
<td>0.20</td>
</tr>
<tr>
<td>Adjusted $R^2$</td>
<td>0.25</td>
<td>0.18</td>
</tr>
<tr>
<td>$F$-value</td>
<td>11.12</td>
<td>11.96</td>
</tr>
<tr>
<td>$P&lt;\alpha$</td>
<td>0.0001</td>
<td>0.0001</td>
</tr>
</tbody>
</table>

$^a y_2 = b_0 + b_1 s + b_2 t + b_3 s t$, where $y_2$ is software product quality.

$^b y_2 = b_0 + b_1 s + b_2 t$. 
Appendix A.

Selected parts of the questionnaire

Dear respondent,

This questionnaire relates to your organizations software applications development. As part of our efforts to investigate software development, your responses are very critical to the success of our study. Please select one development project to respond to this questionnaire. Your responses will be strictly confidential.

Thank you for your cooperation.

A) Please provide the following information about yourself and your organization:

1. The name of your organization. (optional)
2. The primary business of your organization.
3. Number of full-time employees in your Information Systems Department

B) Please provide the following information about the development project you selected:

The size of the development team of this project.

The number of distinct user departments, other than the IS department, involved in the development efforts.

The approximate time spent on the project.

The approximate budget of the development project.

The number of distinct computer programs developed in this project.

The tools used in the development project (check the appropriate answer):

CASE tools.
4GLs.
Database management systems.
Other tools, Specify:

References

Arun Rai is an Associate Professor in the Electronic Commerce Institute, Robinson College of Business, Georgia State University. His research interests include the diffusion, infusion and impacts of information technology, emergent eBusiness models, supply chain management, knowledge management, and management of unstructured processes, such as innovation, product development, and systems development. Arun’s research has been published in journals such as Annals of Operations Research, Accounting, Management and Information Technologies, Communications of the ACM, Decision Sciences, Decision Support Systems, European Journal of Information Systems, Information Systems Journal, Journal of Management Information Systems, Omega and several others. He is an associate editor of MIS Quarterly, e-Services Quarterly, Information Resources Management Journal and a departmental editor of DATA BASE for Advances in Information Systems. Leading corporations, such as A.T. Kearney, Bozell Worldwide, Chrysler, Comdisco, IBM, and Scientific Atlanta, among others, have sponsored his recent research work.

Hindi Al-Hindi is an Assistant Professor of Information Systems at King Saud University. He completed his doctoral degree in MIS from Southern Illinois University at Carbondale in 1996. His research interests focus on management of the systems development process. His previous work has been published in the Proceedings of the Decision Sciences Institute.