Journal of International Technology and Information Management

Volume 14
Issue 1 *Double Issue 1/2*

Article 1

2005

Systems Development Tools and the Relationship to Project Design: Cost and Budget Implications

Gerald V. Post
University of the Pacific

Albert Kagan
Arizona State University

Systems Development Tools and the Relationship to Project Design: Cost and Budget Implications

Gerald V. Post
University of the Pacific

Albert Kagan
Arizona State University

ABSTRACT

An investigation into current system development trends, including a summary of projects, platforms, and tools. The majority of projects get completed on time and within budget, but 14 percent still present problems. The choice of tools appears to be changing to represent the demand for new projects on new platforms—particularly the Internet. CASE tools are used sparingly, and Java is beginning to replace C++ in many projects. Database management systems are critical to almost all projects.

INTRODUCTION

Historically, software development has progressed through several stages. Early programs were written by individuals and small teams. When larger teams began working on large systems, new tools were needed to make it easier to build systems and coordinate the process. Systems development lifecycle methodologies were developed in the 1970s and 1980s to help teams manage the large operations-level projects of the era. With the increasing importance of the windows-based user interface of the 1990s, object oriented programming and eventually object oriented development were created with the goal of developing standardized objects that could be reused in new projects. Similarly, database management systems were created to make it easier to develop and manage common business applications. The use of visual programming tools and RAD development techniques expanded to make it easier to build decision-based systems faster and more efficiently.

During the process of developing new approaches to IT projects, the degree of success has been less than impressive. Cringley (1994) reported that 65% of large-scale systems projects were late or not properly completed. In certain instances these projects were $120 million failures. Many of these systems failed due to bad planning, technological imbalances (developers/programmer issues) and organizational factors. Field (1997) observed that 40% of IS projects fail so that they cost US firms an estimated $100 billion per year. Of the remaining 60% of the projects, one third of these are delayed in time and over expended to the amount of $145 billion per year.

Keil and Robey (1999) argue that US firms must redefine project management activities and change project leadership responsibilities in order to avoid $59 billion in project over runs and $81 billion in project cancellations by US business.

Staw and Ross (1987) observed that IT projects become over budget and delayed because management is reluctant to stop the project. The reticence of management to "pull the plug" on an errant systems project relates to a lack of management awareness of the cost structure of the project or the depth of the overall systems objectives and direction. Oftentimes managers in this circumstance are reluctant to admit failure on their watch.

Numerous studies have discussed the problems of IT project failure, cost over runs and delivery delays (Orli, 1989, Lederer et al, 1990;Betts, 1994; Keil et al 1995). Many IT project difficulties are attributed to four factors described by Keil (1995). These variables are project factors, psychological factors, social factors and organizational factors.

Lederer and Prasad (1995) found that inaccurate cost estimates, relating to IT projects, were caused by poor project management techniques and a lack of system controls. Newman and Sabherwal (1996) provided a set of tactics to reduce IT project failures that included better budgeting activities in the design process and condensed developmental activities using modular deliverables. The direction of these studies is to enhance the management commitment to IT projects with the objective of reducing cost over runs and time delays.

Another view on the issue of IT project design runaways is discussed by Glass (1997). The argument made by Glass is that technological factors are as prominent a cause of system design failures as management factors. New technologies appear to be ahead of the skill level of systems professionals and software developers creating a technological gap between the development tools and the developers. A second technology driven affect is that off the shelf software solutions are not being properly implemented for an efficient and cost effective solution set as when compared to large scale systems development efforts. Cole (1995) reported that the technological factors related to project design were a major reason for system failures, in particular the miss use of packaged software and the ineffective adaptation of these technologies in the project design process.

To facilitate a more accurate approach to IT project design Subramanian and Breslawski (1993) proposed the use of formal modeling techniques to aid in the estimation process of software development. The use of mathematical models should aid in the cost estimation process if these techniques can be included in the IT project management process. Other modeling efforts have questioned the accuracy of cost estimation activities within the software design process (Kusters, van Genuchten and Heemstra, 1990) Zhang et al (2001) suggest that a neural network modeling approach can be used to support software development projects by reducing variability in the design mechanism and more accurately categorize project management information. Smith, Keil and Depledge (2001) used an SEM approach to allow management a better view project status and cost parameters before allowing an IT project to continue in a mode of failure.

As noted in much of the literature there is a need to inculcate new design tools, refined system methods and better project management practices into software design activities. Along with the necessity for changing tools, there is a perception that the business of software design and development is also changing. In particular, large projects or entire departments are being outsourced. Accordingly firms appear to be relying more on purchased software systems solutions (Glass, 1997).

This study examines the impact of new technologies that may be applied to the IT software design process to determine if there will be a cost reduction (or containment) along with a reduction id overall development time. Given the a priori assumption that in the early 1990s, only 26% of IT projects were completed on time and within budget, while 28% of the projects were cancelled, the remaining 46% experienced time and cost overruns (Johnson, 1994 ; Smith, Keil and Depledge, 2001) in excess of $250 billion dollars begs the question of a more efficient process for software development activities. In certain applications it has been more efficient to outsource software projects than use in house development activities (Whitten, Ellis and Casey, 2002).

The expanding importance of Web-based applications has led to even more technologies and “new” development methodologies (Venkatraman, 2004). Firms are faced with the need to rewrite or develop new applications to function in an e-business environment.

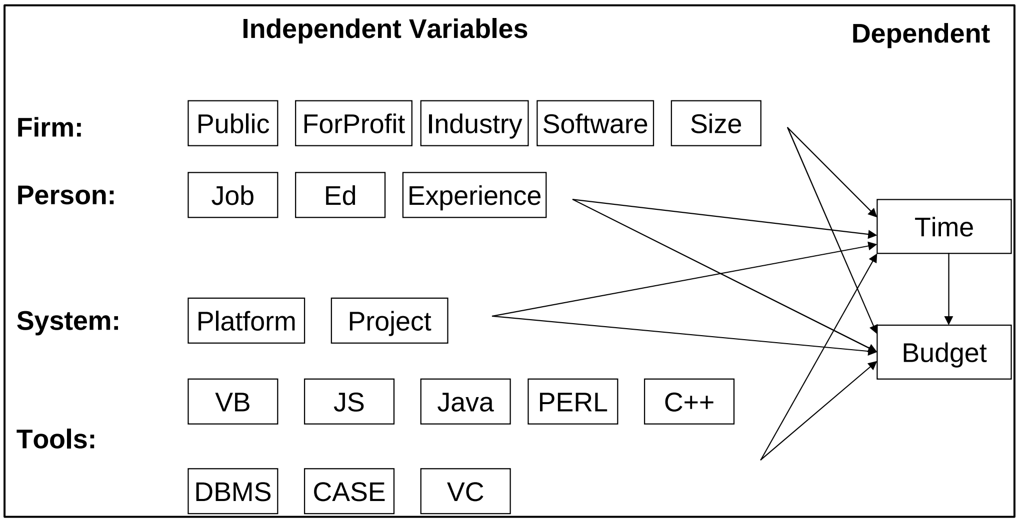
With the expanding number of tools, methodologies, and platforms, many questions remain with respect to the software development and project management process. Ultimately, the most difficult question to answer is also one of the most important: Are the new development tools and methodologies useful in reducing IT system development time or cost escalation? Researchers have investigated these questions from many perspectives. For example, it is possible to examine a single project in extreme detail. But this approach makes it difficult to evaluate multiple technologies and outcomes. The other approach is to evaluate many projects at a lower-level of detail. This method provides a more reliable measure of effects since it covers a wider set of tools and outcomes. On the other hand, it cannot provide a detailed evaluation of all possible effects and influence factors.

MODEL

Dependent Variables

Figure 1 shows the overall structure of the model. The goal is to identify the changes in the dependent variables of budget and time and subsequently the impact of these variables on the systems development process. The dependent variable sets are measured in relative terms using a five-point scale. For example, projects can be completed substantially under budget, under budget, at the budget, over budget, or substantially over budget. The same terms apply to the time variable.

Figure 1: Overall structure of the model



Note that the Time variable will have a direct impact on the Budget process. Projects that are late will often require additional budgetary allocations. Likewise, projects completed ahead of schedule will generally also use less of the allocated budget. This relationship requires inclusion in the final model.

Independent Variables

In theory, several items could affect the results. These items are grouped into four categories: (1) the firm, (2) the person involved in the project, (3) the type of system being developed, and (4) the tools used. These items are a modification of the 4 factors that affect systems development projects presented by Staw and Ross (1987) and Keil (1995). Table 2 shows the connection between the four categories in this study with the Staw and Ross classification.

Firms are grouped into industries defined by standard classifications (e.g., the top level of the NAICS). Firms are also identified as "for profit" and whether or not they are publicly traded. The size of the firm is also a potentially important issue. Larger firms can have more complex projects, which require larger staffs and possibly more experience. It is not clear which effect will dominate, but this issue needs to be evaluated. Given the expanding importance of independent software vendors, it is important to identify whether the project being evaluated is developed within a software specialist firm or within a traditional (internally developed) business unit.

Various measures are typically made on individuals. Common elements include education, experience, and the type of job classification. In models of this type, jobs reduce to levels ranging from programmer to analyst to systems engineer to manager. The job variable was coded along that range. Hence, in addition to testing for differences in job effects (evaluated by significance), it can also provide an estimation of the effect of higher-levels of management (the sign of coefficients).

The type of system was categorized according to the platform and the descriptions of the projects. Platforms ranged from small PC-based to client-server systems, to applications for the Internet and wireless systems. Projects included items such as systems programming, transactions and accounting systems, decision support, enterprise resource planning (ERP), and e-commerce applications.

Projects often use many tools, so the model uses binary variables to identify the use of a variety of common tools, including Visual Basic (VB), JavaScript (JS), Java, PERL, C++. Three additional development tools were evaluated because of their importance in the methodology literature and potential impact on development efficiency: database management systems (DBMS), computer-aided software engineering (CASE), and version control systems (VC).

Additional Variables

In some situations, it would be necessary to consider an additional dependent variable category for projects that were abandoned. However, these projects would often fall into the categories of “substantially over” budget and time (Cole, 1995). Additionally, respondents in this study did not report projects in that category.

In a detailed project analysis, it would also be possible to evaluate the skills and experience of each team member as well as the project manager. However, most prior research has shown that these measures do not significantly affect most projects—generally on the grounds that most workers meet at least minimal requirements to complete the job. However Jiang, Klein and Chen, 2001 argue that project performance and outcome are related to project management effectiveness. Additionally, the measures of the respondent can serve as proxy variables for the team.

Prior studies have also shown that some management variables play a role in successful project completion (Keil, Mann and Rai, 2000). Notably, top management support has often been reported as an important factor. Additionally, behavioral characteristics within an organization could influence productivity. However, these variables are difficult to evaluate across companies. The effects are also mitigated in this study, because most respondents provided data on three separate projects.

Factors

With detailed surveys of a small number of projects, it is possible to create multiple measures of various factors. Latent variable analysis can then be used to create a measurement model that identifies unobservable factors and compensates for errors in measurement. However, latent variables are best measured with continuous variables. While categorical variables can be used, the large number of them in this study makes it difficult in practice to identify separate factors. For example, it would be tempting to create separate factors for the four categories (firm, person, system, and tools). However, these factors would be too broad to be measurable. More importantly, combining all of the effects into four factors would hide the importance of the individual effects. For example, it is important to evaluate the detailed effects of each tool—as opposed to a single effect of a tool factor. Consequently, the model is best evaluated as individual effects instead of aggregated factors.

Effects and Relationships

The primary objective of this study is to evaluate the effect of the independent variables on the dependent variables. The degree of significance, sign direction, and the relative strength of these effects can provide information to guide management in evaluating and adopting tools to improve software development efficiency.

As with any study in an uncontrolled environment, many interactions exist among the variables. These relationships are important both from a management perspective and must be controlled from a statistical perspective. For example, it is to be expected that many of the independent variables are correlated and affect the results indirectly through other variables, along with the direct effects. These effects are measured as correlations among the variables. Due to the complexity of the interactions it is not possible to show the correlations in Figure 1. Only the direct effects are summarized—as arrows.

Survey

An objective of this study was to obtain a relatively broad picture of systems development methodologies in the context of time and budget applications. Consequently, lists of developers and managers were generated from businesses around the United States. These lists were developed from system professional user and discussion groups, general business contacts and specific referrals. Obtaining complete survey responses was important but challenging. To make the survey easy to use, it was conducted online over the Internet. Randomly selected respondents were provided the URL. Duplicate and incomplete responses were discarded. To increase the number of responses, the survey was ultimately conducted in three waves in slightly under a year. Responses in the three waves are identified by a collection variable (Survey Set).

Collection of survey responses over multiple time sets provides a basic test of non-response bias. By comparing means from the three sets it is possible to identify differences in the groups. The three sets generate three t-tests: set 1 versus set 2, set 1 versus set 3, and set 2 versus set 3. The samples were compared on the basis of firm, individual measures, and tool usage. Almost none of the variables are significantly different between the first and second sets. On the other hand, many of the firm and personal measures are different in the first-and-third and second-and-third comparisons. The implications are that there was essentially no non-response bias in the initial set of subjects. Since the third set was generated from new lists, the differences are attributable to the slightly different population. In particular, the third set tends to represent substantially larger firms (18,000 versus 2,000 employees). Respondent personal differences with the third set are not generally significant. Also, tool and platform differences are similar in all three sets.

Respondents and Firms

Software developers and their managers were the primary targets of this study. The basic characteristics of the 136 respondents (which includes 371 project evaluations) are shown in Table 1. Respondent demographics indicate that they have considerable experience in designing and building software applications. Education and experience attributes were coded to measure years, so Table 1 indicates six years of programming and development experience, with slightly over three years in the current job. The average respondent worked for four different firms, providing a fairly wide level of experience in different situations. The education average of 4.74 indicates that many of the respondents have master's degrees. The jobs were coded from zero to 8 as: unspecified (2), programmer (36), programmer/analyst (25), analyst (16), engineer (2), team leader (24), MIS manager (16), Manager/non-IS (8), and CIO/CTO (7). The bulk of the respondents were actual developers (programmers, analysts, and team leaders).

Table 1: Attributes of the individuals.

Variable	Mean	Standard Deviation
Education (years past high school)	4.74	1.21
Current Experience (years in current job)	3.21	2.38
Overall Experience (years)	6.04	3.70
Experience with firms (number of companies worked for)	4.05	2.28
Department Size (number of employees at site)	38.41	56.64
Job: Programmer Analyst Team Leader Other	36 41 24 35	

Table 2 illustrates the diversity of companies represented by the respondents. For example, the high variance on the number of employees and sales indicates that responses were obtained from a wide variety of companies. Almost half of the firms are publicly traded. Almost three-fourths of the responses came from firms with a focus on software development (e.g., independent software vendors).

Table 2: Attributes of the firms.

Variable	Mean	Standard Deviation
Public	0.46	0.50
For Profit	0.92	0.27
Employees	7148	17,214
Sales ($ Millions)	1152	2135
Software focus	0.72	0.45

Projects

Respondents were asked to evaluate up to three recent projects they worked on. Almost all of the respondents did evaluate three projects. The types of projects are summarized in Table 3. Respondents were asked to classify projects in terms of the platform (e.g., PC, client/server, and Internet). They also identified the type of project (e.g., ranging from systems to decision support to e-commerce). Additionally, respondents indicated the primary tools used on the project. In this category, they could select multiple tools from a list. Many projects used more than one tool. This selection does not present a problem for the summary statistics. However, in the structural equation model, the variables were recoded as binary to indicate the presence or absence of a particular tool on a project.

Table 3: Project attributes. Projects often used more than one tool.

Attribute	Count
Platform	
Unspecified	2
PC	19
Server/Mainframe	23
Client/Server	53
Internet	32
Wireless	7
Project	
Unspecified	3
Systems/programming	22
Transactions/accounting/HRM	31
Process control	12
Decision support	12
Financial management	6
Enterprise resource planning	13
CRM/Marketing management	12
e-Commerce: Retail/B2C	17
e-Commerce:B2B/EDI	8
Tools	
Visual Basic (VB)	110
JavaScript (JS)	25
Java	94
PERL	31
C++	122
DBMS	163
CASE	37
Version Control (VC)	31

Lastly, respondents were asked to evaluate the basic efficiency of the project: Was the project completed on time and on budget? For both time and budget, they indicated the outcome on a five-point scale: substantially early/under, early/under, on time/budget, late/over, or substantially late/over. The summary results are shown in Table 4. Interestingly, the majority of projects were completed on-time and within the specified budget. None of the projects were identified as substantially over time or budget. On average, the projects were slightly under time (-0.16) and budget (-0.18). The summary results indicate that it is rare for projects to fall into the substantially under and substantially over categories. The 95 percent confidence interval is approximately (-1.5 to 1.2). Notice the difference between these results and the values reported by earlier studies. Part of the difference could be attributable to more recent tools or other improvements in development and project management techniques. Some of the effect might be due to the narrower vision of development professionals versus top executives.

Table 4: Project outcome summary.

Outcome	Time (count of projects)	Budget (count)
Substantially Under	8	7
Under	95	106
As specified	216	205
Over	52	53
Substantially Over	0	0
Mean	-0.16	-0.18
Standard Deviation	0.68	0.69

RESULTS

Model Evaluation

Several methods exist to evaluate the fit of structural equation models. Two of the more common measures are the Chi-square statistic and the root-mean-square-error (RMSEA). In this model, the Chi-square statistic is relatively low at 19.7 with 23 degrees of freedom, yielding a probability level of 0.658; indicating the model is reasonable. The RMSEA is essentially zero (to 3 decimal places). The other statistics are equally strong, indicating that the model is acceptable.

One other measure of the model performance is to examine the squared multiple correlation coefficients (analog to the regression R-square terms) on each equation. The value on the Time equation is 0.15, and the Budget equation has a value of 0.45. The value for the Time equation is a little low. The result indicates that factors outside those measured will affect the ability to bring a project in on time. The value for the Budget equation is better (but note that the Budget equation depends on the Time variable). Most likely, organizational and management factors that were not measured in this study will have a strong effect on the development team's efficiency.

While it would be useful to identify all of the major variables, it is probably not possible to do so within one study. From a statistical perspective, as long as the overall error values are low enough to hold down the unspecified error terms, the overall model is acceptable. If the unspecified error is too high, then it would not be possible to identify any significant individual variables. As discussed in the next section, that situation does not arise here, so the model is acceptable for the purposes of this study.

In theory, two latent variables could have been estimated from the data: one for experience and one for company size. When these latent variables were estimated in isolation from the rest of the model, the results were mediocre. If they are included within the overall model, the weaknesses in these estimates lead to instability of the model. In particular, it is difficult to get consistent measures from respondents on firm size. The common measures of number of employees and sales tend to conflict—partly because respondents do not seem to know the true values, partly because the relationship between number of employees and sales depends on the type of industry. Since the absolute magnitude of the relationship in both cases is not critical, any one of the measures can be used as a proxy variable and the results are almost identical. Consequently, the number of employees was chosen to estimate the size of the firm and the number of firms an individual worked for as the most accurate proxy for a respondent's experience.

RELATIONSHIPS

Primary measures of the relationships are presented in Table 5, which shows the values of the path coefficients for the Time and Budget equations. In viewing the Time equation notice that six of the coefficients are significantly different from zero. Because Time is a significant factor in the Budget equation, these six variables will also affect the Budget results. Three other variables are significant in the Budget equation. These variables are important because they indicate items that directly affect the efficiency of the software development process.

Table 5: Path coefficient results from structural equation modeling.

Variable	Time		Budget	
	Coefficient	Ratio	Coefficient	Ratio
Time			0.609	14.212**
Public	0.084	0.991	-0.125	-1.753
ForProfit	-0.063	0.147	0.011	0.089
Industry	0.034	2.198*	0.007	0.552
Software	-0.005	-0.052	-0.066	-0.797
Size (Emps)	0.000	0.111	0.000	0.652
Dept (size)	0.001	0.755	0.002	2.210*
Job	0.016	1.022	0.001	0.068
Education	0.086	2.775 **	-0.013	-0.491
Experience (Firms)	0.014	0.854	0.031	2.223*
Platform	0.081	2.475*	-0.026	-0.956
Project	0.019	1.518	-0.006	-0.573
Visual Basic	-0.226	-2.690**	0.079	1.125
JavaScript	0.125	0.897	0.041	0.359
Java	-0.258	-3.028**	-0.009	-0.128
PERL	-0.233	-1.910	-0.050	-0.500
C++	-0.121	-1.644	-0.075	-1.228
DBMS	-0.287	-4.070**	-0.129	-2.176*
CASE	-0.119	-1.876	-0.188	-1.876
Version Control	0.044	0.333	0.095	0.860
Survey Set	-0.072	-1.464	-0.018	-0.161

* One asterisk represents significance of the critical ratio at a 5 percent level, two asterisks signify 1 percent.

As expected the Time variable has a highly significant effect on the Budget. This result indicates that projects that are late will also tend to drive the project over budget. The Time/Budget interaction supports prior research as the metric for non completion of software projects.

It is also useful to note that the Survey Set variable did not have a significant effect on either the Time or Budget variables. This result affirms the test of the basic means that showed only a few differences in the respondent characteristics between the three applications of the survey. In other words non response bias was not present.

Environmental Elements

The Industry variable is one of the few firm and personal attributes that appear to make a difference. The sign of the coefficient is not important since the industries were randomly coded. The fact that it is significant indicates that some industries are inherently more or less efficient than others in developing software applications. Looking at the means for the Time variable within the various industries, three stand out from the others. The mean response in educational institutions was the smallest (most negative) value at -0.31. The mean response in general manufacturing was the highest at 0.10. The most likely explanation for the difference is the greater challenges in some industries—such as manufacturing. Another possibility is that perhaps some industries have less resources, or lower-qualified developers. But, since educational institutions performed better than the others, and they tend to have the least resources and lower salaries, this effect seems unlikely. The manufacturing firms are probably affected by the fact that they had the respondents with the least experience (average of 3.5 total years versus 6 years overall). A secondary effect can be the changing nature of the systems project, in that certain industries have had more instability with respect to user needs and definition. The other important industry was software development with a mean value of -0.20 (the second lowest value). This value makes sense given the industry's needs and experience. If software vendors were less efficient than other firms, it would be difficult to be profitable.

Consider the Education variable, which is the other personal factor that has a significant effect on the Time equation. Interestingly, it is positive, meaning that respondents with more education reported more delays in development projects. The negative interpretation of this coefficient would be that developers with more education are less efficient. But there are several alternative explanations. Possibly developers with more education are better able to recognize and judge delays (or are more critical of them). Or, perhaps more highly educated respondents were called in to work on more difficult projects or projects that were already in trouble.

In terms of the Budget equation, experience again shows a positive relationship—this time in terms of the number of firms the respondent has worked for. Fortunately, the magnitude of the coefficient is relatively small at 0.031, so the practical impact is somewhat limited. Again, there could be a relatively innocent interpretation of this effect. It is possible that employees who experienced more difficulties with projects changed jobs more often.

A more important effect on the Budget equation comes from the size of the department the respondent works in. The positive value indicates that larger departments tended to result in going over budget on projects. While the coefficient seems small in magnitude (0.002), remember that it is multiplied by the number of employees. With an average of about 38 employees, this variable would affect average firm by 0.076. More importantly, it indicates that as departments get larger the total effect increases. This result affirms part of Brooks' long-known *Mythical Man Month* (1975) statement that adding developers to a project increases the budget costs and further delays the project. This result is also consistent with the findings of Staw and Ross (1987), Cole (1995) and Keil (1995). In the sample, firms with 3 employees had an average budget result of -0.28, while departments of 150 (or more) people had a budget result of 0.11. However, departments with 75 employees had the lowest average at -0.32. So, there appears to be a discrete jump effect. Up to a value of 75-100 employees the budget effect is minimal. Beyond that, it becomes critical.

Project Characteristics

The strength of the Platform variable indicates that it can be inherently more difficult to develop software for some platforms. Notice that the coefficient is positive, and that the platforms were coded in an approximately increasing level of difficulty (e.g. PC to client/server to wireless). Examining the means for Time within each Platform shows that the PC platform had the lowest mean at -0.39, client server the second-lowest mean at -0.21, and wireless the highest mean at 0.09. Given the fact that wireless is the most recent platform, with minimal tools and limited experience by developers, it is to be expected that it presented the greatest challenges in terms of development time. Similarly, standalone PC applications are generally smaller, less complex, and have many powerful development tools.

Tools

Some of the most important results of the study arise by looking at the significant coefficients for the various tools. Three of them stand out from the others: Visual Basic, Java, and database management systems. In all three cases, the tools have significant negative coefficients, and the coefficients have relatively high magnitudes (-0.226, -0.258, and -0.287). The interpretation is that each tool substantially reduces the development time of the project. If a project could use all three at the same time, the total effect is -0.77; or almost enough to move a project from on-time to early. Note that only five projects in the sample used all three tools. However for those projects, the mean Time was -0.80, compared to -0.15 in the other cases. So the strong time-savings effect is demonstrated in those cases.

A few other tools are not technically significant, but the coefficients are relatively strong negative values: PERL, C++, and computer-aided software engineering (CASE) tools. With a larger study, or measures of other management variables, it is possible that these values would become significant. The only tool that shows the potential for being problematic is JavaScript. Again, the value is not statistically significant, but the coefficient of positive 0.125 indicates that JavaScript might be more difficult to work with than the other tools. Since only 25 projects in this study used JavaScript, it would be a good candidate for additional study—particularly with the introduction of several JavaScript generating tools.

Notice that a DBMS is the only tool that also has an additional significant effect in the Budget equation. Again, the value is negative; indicating that using a DBMS in the project makes it more likely the project will be completed under budget or substantially under budget. Keep in mind that variables that are significant in the Time equation will also indirectly reduce the budget costs. But the DBMS has an additional direct effect. An examination of the means of the Time and Budget variables reveals a strong difference in projects that used a DBMS and those that did not. Projects without a DBMS had means for Time and Budget of -0.058 and -0.063 respectively. Projects using a DBMS had Time and Budget means of -0.288 and -0.331 respectively.

Correlations

Many of the independent variables are correlated with each other. These effects were measured and statistically compensated for with the SEM approach. Because of the large number of effects it is not possible to present all of them. Table 6 presents some of the correlations—notably those involving the tools and project variables. Notice that Visual Basic (VB) is significantly negatively correlated with most of the other tools. That is, projects that use VB tend not to use any of the other tools. As expected by Web developers, JavaScript is positively correlated with PERL. Java is negatively correlated with C++ because it is largely seen as a replacement in the object-oriented world. CASE tools and version control (VC) are strongly positively correlated, showing that they are often used together.

Table 6: Selected correlation coefficients.

	Platform	Project	VB	JS	Java	PERL	C++	DBMS	CASE
Project	0.23**								
VB	0.01	-0.11*							
JS	0.06	0.15**	-0.15**						
Java	0.18**	0.30**	-0.31**	0.09					
PERL	0.06	-0.01	-0.06	0.11*	0.07				
C++	-0.09	0.05	-0.15**	-0.10	-0.14**	-0.09			
DBMS	-0.01	0.16**	-0.15**	-0.15*	0.07	-0.09	0.09		
CASE	-0.02	-0.04	-0.20**	0.09	-0.02	0.04	0.01	-0.08	
VC	-0.10	-0.05	-0.17**	0.07	-0.04	0.06	0.03	-0.05	0.35**

* One asterisk indicates significance at 5 percent level, two at one percent level.

In terms of platforms, the correlation between Platform and Project indicate that developers consistently choose certain platforms for specific projects. In general, the choice of tools within a platform is neutral—except for Java. The means indicate that Java is often used for Internet and wireless development.

With respect to projects, Visual Basic, JavaScript, Java, and DBMS tend to be correlated with the choice of projects. Visual Basic is most often used for decision support projects (in over fifty percent of the 38 projects). JavaScript is used most often for e-commerce projects—both B2C and B2B, but in only 14 and 11 percent of the 64 and 54 projects respectively. Java is used most in B2B projects (61 percent of 54 projects) and B2C (38 percent of 64 projects). It was also in 25 percent of the 40 systems projects. The issue of using a DBMS is slightly different from the others. It is heavily used in most application projects (50 to 60 percent of the time). But systems projects used a DBMS only 20 percent of the time.

Most of these correlations agree with the common knowledge in the industry. Besides the statistical correction factor, the main interest in the statistics is that they provide a measurement and a significance test of the various industry practices.

CONCLUSION

As noted, 14 percent of the systems projects were over budget and late. The type of industry plays a role in development efficiency, with projects at manufacturing firms more likely to be late and over budget. Newer platforms—notably wireless—also caused more problems in completing projects on time. This effect is likely due to the limited experience and scarcity of development tools keyed to the new platforms. The study also clearly showed that large development organizations (150 or more developers) were more likely to run over budget on projects.

The strongest results of the study are that certain tools make it significantly more likely that projects will be completed on time and within budget. A database management system is the most important tool of the group—having a strong impact on both the time and budget variables. Visual Basic and Java were the other tools that had significant effects at reducing development time (and budgets indirectly).

Industry implications are straightforward: applications can be developed more efficiently by small to medium-size development teams using a DBMS and Visual Basic or Java programming tools. These recommendations apply particularly well to decision-support projects. Projects developed for newer platforms without these types of tools available should expect to require more time. Be careful with this last statement. It does not mean the projects should be avoided. It simply means that managers and developers should plan for more complications.

MANAGERIAL IMPLICATIONS

Many studies have discussed the problems of software project design and the reasons for runaway projects. The factors contributing to project escalation and non completion are described by various authors (Staw and Ross, 1987; Keil, 1995; Keil and Mann 1997). These factors relate to the taxonomy structure in this paper that describes firm level issues, tool driven factors, developer (person) skill and experience and the type of system being developed.

Some modern tools appear to be useful at avoiding budget and time overruns. Applications built from a DBMS and tools like Visual Basic and Java that insulate the developer from common problems seem to minimize problems. However, some industries need to be more careful than others. Managers of larger projects particularly those involving larger departments, need to be more careful in terms of budget overruns.

APPENDIX

Survey Instrument

The actual instrument runs as a Web form, so the layout has been modified for printing.

Ownership: Public / Private

Nature: For Profit / Not for Profit

Industry: Open-ended and classified by researchers

Employees: 1-9 / 10-199 / 200-999 / 10,000-49,999 / 50,000+

Sales 1-99,999 / 100,000-1M / 1.1M-10M / 10.1M-100M / 101M-1B / 1.1B-10B / >10B

Org. Size

Focus Software development focus Yes / No

Job Programmer / Programmer-Analyst / Analyst / Engineer / Team Leader / MIS Mgr / Management-Non IS / CIO-CTO / other

Education High School / Two-year / Bachelors / Masters

Exper. in current <2 years / 2.1-5 / 5.1-7 / 7.1-10 / >10

Exper. overall <2 years / 2.1-5 / 5.1-7 / 7.1-10 / >10

Companies 1 / 2-4 / 5-7 / 8-10 / >10

	Project 1	Project 2	Project 3
Project type	Unspecified / Systems-Programming / Transactions-Accounting-HRM / Process control / Decision support / Financial management / ERP / CRM-Marketing		

	management / EC: Retail B2C / B2B-EDI		
Platform	PC / Server-Mainframe / Client-Server / Internet / Wireless		
Tools	Unspecified / Visual Basic / JavaScript / Java / PERL / C++ / DBMS / CASE / Version Control		
Time	Early / On time / Late / Extremely late		
Budget	Under / On budget / Over / Extremely over		

REFERENCES

Betts, M. (1993). How to Get 'Runaway Projects' on Track, *Computerworld* 27(1l), 95.

Brooks, F.P. (1995) *The Mythical Man-Month, Anniversary Edition*, Addison-Wesley. [Or see the original 1975 version.]

Cafasso, R.. (1994). Few IS Projects Come in on Time, on Budget. *Computerworld* 28(50), 20.

Cole, A., (1995). Runaway Projects: Cause and Effects. *Software World* 26(3), 3-5.

Cringely, R.X.. (1994). When Disaster Strikes IS: How to Forfeit Millions in Exchange for Nothing, *Forbes (ASAP Supplement)* 154(5), 60-64.

Field, T. (1997). When Bad Things Happen to Good Projects. *CIO* 1l(2), 54-62.

Glass, R.L. (1997). Software Runaways - Some Surprising Findings, *The DATABASE for Advances in Information Systems* 28(3), 16-19.

Jiang, J. J., G. Klein, and H. G. Chen. (2001). The Relative Influence of IS Project Implementation Policies and Project Leadership on Eventual Outcomes, *Project Management Journal* 22(3), 49-55.

Johnson, J. (1995). Chaos: The Dollar Drain of IT Project Failures, *Project Management Journal* 2(1), 41-47.

Keil, M. (1995a). Identifying and Preventing Runaway Systems Projects. *American Programmer* (March), 16-22.

Keil, M. (1995b). Pulling the Plug: Software Project Management and the Problem of Project Escalation, *MIS Quarterly* 19(4), 421-447.

Keil, M. and D. Robey. (1999). Turning Around Troubled Software Projects: An Exploratory Study of the Deescalation of Commitment to Failing Courses of Action, *Journal of Management Information Systems* 15 (4), 63-87.

Keil, M., J. Mann and A. Rai. (2001). Why Software Projects Escalate: An Empirical Analysis and Test of Four Theoretical Models, *MIS Quarterly* 24(4), 631-664.

Kusters, R. J., M. J. van Genuchten, and F. J. Heemstra. (1990). Are Software Cost-Estimation Models Accurate? *Information and Software Technology* 32(3), 187-190.

Lederer, A. L., R. Mirani, B. S. Neo, C. Pollard, J. Prasad and K. Samamurthy. (1990). Information System Cost Estimating: A Management Perspective, *MIS Quarterly* 14, 159-176.

Lederer, A.L. and J.Prasad. (1995). Causes of Inaccurate Software Development Cost Estimates. *Journal of Systems Software* 31(2), 125-134.

Newman, M. and R. Sabherwal, (1996). Determinants of Commitment to Information System Development: A Longitudinal Investigation, *MIS Quarterly*, 20(l), 23-54.

Orli, R.J. (1989). Battling Project Escalation, *Computerworld* 23(14), 64,66.

Smith H.J., M.Keil and G. Depledge. (2001). Keeping Mum as the Project Goes Under: Toward an Explanatory Model, *Journal of Management Information Systems* 18(2), 189-227.

Subramanian, G. H. and S. Breslawski. (1993). Dimensionality Reduction in Software Development Effort Estimation, *Journal of Systems and Software* 21, 187-196.

Staw, B.M. and J. Ross. (1987). Knowing When to Pull the Plug, *Harvard Business Review* 65(2), 68-74.

Venkatraman, S. S. (2004). Web-Services–The Next Evolutionary Stage Of E-Business**.** *Journal of International Technology and Information Management* 13(2), 111-122.

Whitten, *G.* D.T. S. Ellis, and K. M.Casey, K. M. (2002). The Impact of Information Technology Outsourcing on Firm Profitability Measures**,** *Journal of International Technology and Information Management* 11(2), 9-18.

Zhang G.P.; M. Keil, A. Rai and J. Mann. (2003). Predicting Information Technology Project Escalation: A Neural Network Approach, *European Journal of Operational Research* 146 (1), 115-129.