8-15-1997

Software Project Management: The Relationships Among Effort, Change and Time,or A First Attempt at Preventing Scope Creep

Evelyn Barry
Carnegie Mellon University

Tridas Mukhopadhyay
Carnegie Mellon University

Sandra A. Slaughter
Carnegie Mellon University

Software Project Management: The Relationships Among Effort, Change and Time, or A First Attempt at Preventing Scope Creep

Evelyn Barry
Tridas Mukhopadhyay
Sandra A. Slaughter
Graduate School of Industrial Administration, Carnegie Mellon University
Pittsburgh, PA 15213
phone: (412) 268-3681, fax: (412) 268-7064

INTRODUCTION

We examine a chronic problem in software development: why do software projects finish over budget? Prior work focuses on developing better effort estimation models by continued attempts to correct specification and measurement errors. These models all assume a static project environment. However, most projects continue to be under-estimated.

We take a different approach. We suggest that project size metrics and project characteristics are sufficient to estimate project effort for the *original* project. This implies that the difference between required project effort and estimated project effort can be explained by new project requirements that occur after estimation.

We hypothesize that the project grows with time as the business and technological environments force changes in project requirements. Our empirical test of this hypothesis supports our claim. By comparing the growing project size with budgetary limits we can determine a maximal project duration. We examine a number of software development projects in a government organization and find that there is a maximal project duration or project scope beyond which projects will grow out of control and over budget.

THEORY

After system requirements are established, the amount of required effort is estimated for a software project. When the estimate is approved by the stakeholders, the project is started. Project duration is the elapsed time between project start and project completion. As time passes, the project environment changes. There may be changes within the project team and also changes in a multi-project environment. These changes are under the control of the project manager. However, software development work may also be affected by many changes in the surrounding business and technological environment that are beyond the control of the project manager. For example, business changes will change the users' requirements, which in turn will force changes in software system requirements. Theoretically, as changes are required, the project manager should re-estimate the project. However, given the constancy of change, time spent re-estimating may exceed time spent on actual software development. Once development work starts in earnest, many project leaders and project managers handle this situation by holding system specifications constant and allowing only "essential" changes to be accommodated. The longer a project takes, the more futile it becomes to pretend that no changes in requirements have occurred. Often the original estimate is no longer valid.

Thus, it is important to recognize the effect of environmental changes on software requirements. In our model of software development effort we make the following assumptions about those changes:

- changes beyond the control of project managers are relevant to software requirements
- changes are continuous and affect all elements of a project equally
- changes compound, i.e. changes occurring today build on those which occurred yesterday

Because change occurs over time, we are concerned with the duration of a software development project and its relationship with required project effort.

We suggest that as project duration increases, more changes occur in the surrounding environment which lead to changes in user requirements. These changes compound with the passage of time and cause the software project to grow exponentially. If A is the original software project, and A' is the completed project, then $A' = A(e^{\ *duration})$, where is the rate of environmental change.

This has two implications: 1) there is a maximal duration or scope for projects beyond which they will grow out of control, 2) this recommended maximal duration can be determined by budgetary limits and the rate of environmental change. (see figure 1)

EMPIRICAL STUDY

An empirical study was conducted to test our hypothesis. The research site selected is a small application development and maintenance group in a state government agency. Data were collected on 18 projects completed between 1989 and 1994. This site provided a natural control for some project factors which facilitated our examination. Particular site characteristics were:

1. All projects had team size of only one or two programmers, with a minimum of personnel turnover on both the programmer and user staffs.
2. All programmers on the team had at least 10 years professional experience at the start of the data collection period.
3. The projects were all enhancements to COBOL/VSAM/CICS applications in the same strategic business unit of the agency.

We formulated and estimated the following mixed model for software development effort: $\ln(\text{Actual Effort}) = {}_0 + {}_1(\text{duration}) + {}_2\ln(\text{project size}) + {}_3(\text{impcd}) + {}_4(\text{pgmr skill})$. (see table 1 for variable definitions) We verified the log transformation for project effort with a Box-Cox test. We tested both a simple model with one independent variable, and a complex model with four independent variables. The results for both models are listed in

Table 2. In both estimated models, the coefficient for project duration was positive and statistically significant implying that project duration has a significant and positive effect on project effort. Based on the log transformation, we can conclude that the relationship between project effort and duration is exponential for this set of projects.

The site specific results obtained in this research confirm a non-linear relationship between project duration and project effort. In addition, the technique for determining the duration limit which exceeds spending limits was applied to the data. For this site, and assuming a 10% cost overrun spending limit, these data indicate that the projects studied should not last longer than 141 calendar days or changes in requirements will cause them to exceed budget. This time limit can be compared on an order of magnitude with the efficient project size of 235 calendar days obtained by Banker and Slaughter when studying a commercial organization. (Banker and Slaughter, 1996).

IMPLICATION AND CONCLUSION

This work has expanded prior research for effort estimation models by including a growth function which accounts for a changing project environment. We suggest that a new approach should be taken to look for new explanations in effort prediction errors beyond the traditional search for specification and measurement errors.

Project managers who are able to anticipate the rate of change in their specific business environment can use that information and their estimates of effort requirements to improve their ability to meet project targets. Using this exponential model, managers can anticipate the time when project effort is likely to grow beyond spending limits. Project managers could also use this time limit to check anticipated project schedule to see if the project scope is too broad to be completed within their planning horizon. This permits projects to be scoped so they can be completed in reasonable time spans, thus preventing scope creep.

The current practice of time-boxing sets a 60 day time limit for project assignments. These results imply that a contingency approach should be tried. The time limit should be calculated depending on budgetary limits and the rate of change in the specific business environment under development.

This research has demonstrated a significant, positive and non-linear relationship between project effort and project duration. Further study would help to generalize these results and investigate other dynamic factors such as learning effects and inter-project correlation which could also affect this model.

References furnished upon request.

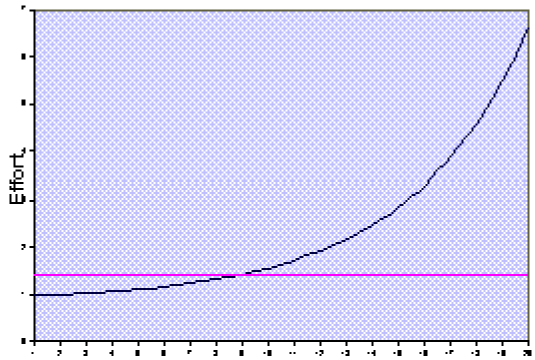

Figure 1: Project Effort vs. Project Duration

Variable	*Definition*
Duration	Calendar days between start and completion of project
Impcd	Binary variable for inclusion of imported code used in project; yes or no
Pgmr skill	Binary variable to classify project staff members as having junior or senior skill level
Project size	Metric to determine relative size of project requirements; measured by number of programs

Table 1: Variable Definitions

Variable	*Coefficients Simple Model:* *Adj.*R^2*=.63748* *F = 15.95 ** *	*Complex Model:* *Adj.*R^2*=.75512* *F = 14.11 ** *
duration	.001167 *	.000678 *
ln(project size)	.458726 *	.581310 *
impcd		- .977291 *
pgmr skill		.358569 *
constant	4.280909 *	5.187859 *

** significant with p-value < .001 * one-sided t-test significant at .10 level

Table 2: Regressions on ln(project effort)