A computational econometric modeling technique for the parametric simulation of effort and duration estimation as applied to the development of software systems

By

Peter J. Morales

Submitted in partial fulfillment
of requirements for the degree of
Doctor of Professional Studies in Computing

at

The School of Computer Science and Information Systems

Pace University

July 25, 2011

PREVIEW

UMI Number: 3491863

All rights reserved

INFORMATION TO ALL USERS
The quality of this reproduction is dependent on the quality of the copy submitted.

In the unlikely event that the author did not send a complete manuscript
and there are missing pages, these will be noted. Also, if material had to be removed,
a note will indicate the deletion.

UMI 3491863

Copyright 2011 by ProQuest LLC.

All rights reserved. This edition of the work is protected against
unauthorized copying under Title 17, United States Code.

PREVIEW

ProQuest LLC.
789 East Eisenhower Parkway
P.O. Box 1346
Ann Arbor, MI 48106 - 1346

We hereby certify that this dissertation, submitted by Peter J. Morales, satisfies the dissertation requirements for the degree of Doctor of Professional Studies in Computing and has been approved.

_____ - _____

Dr. Fred Grossman Date
Chairperson of Dissertation Committee

School of Computer Science and Information Systems
Pace University 2011

PREVIEW

An Abstract

A computational econometric modeling technique for the parametric simulation of effort and duration estimation as applied to the development of software systems

By
Peter J. Morales

Submitted in partial fulfillment
of the requirements for the degree of
Doctor of Professional Studies in Computing

July 2011

Abstract

Software development continues to be challenged by projects that are often late or over budget. In this dissertation we review the relevant traditional and agile approaches to the governance control and management of software projects. We explore reasons for common disconnects between initial estimates and the final resulting cost, duration and effort metrics for software projects. We propose that improving the accuracy of initial estimation would result in a better match between expectations and results in software development. Finally we introduce a detailed parametric modeling and simulation approach for developing early effort and schedule estimates for a software project.

The detailed parametric model for software project effort and schedule estimation proposed here incorporates granular parametric modeling techniques coupled with process simulation and analysis techniques in order to address a broader range of project scenarios. We propose that simpler versions of the parametric basis formulas can be employed and that process analysis can improve the utility of the effort and schedule estimation for real world projects thus improving the management of risk. The model also incorporates a re-work feedback loop that models the disruptive behavior of bugs. We propose that the 'transfer functions' used in this model provide a more granular approach that can be tuned locally for an on-going project as work begins and provides more actionable and immediate feedback. We further propose that this detailed parametric model deals with real world problems such as changing requirements and variations in resource availability in ways that traditional parametric models cannot.

PREVIEW

While the model developed here is based on a specific software development life cycle (Unified Process) used in the case study project for a Fortune 500 company, it can be applied to any defined methodology that can be represented by stages of work. These software development life cycle stages of work can be linked sequentially or in parallel to complete a given final work product or products. In fact, the modeling and simulation methodology developed here can be applied to any business process life cycle, not just software development.

PREVIEW

Dedication

This dissertation is dedicated to my wonderful and loving family; my wife Karen, my son Christopher and my daughter Katie. Without their support and encouragement I would not have been able to complete this work.

I would also like to thank my father Pedro, who as an engineer taught me a love of mathematics and science, and my mother Anna, who as a teacher and poet taught me the beauty of words and their ability to move people and shape action. While they are not with us in body, I hope they can observe and enjoy the fruits of their labor.

PREVIEW

PREVIEW

PREVIEW

Table of Contents

PREVIEW

PREVIEW

PREVIEW

PREVIEW

PREVIEW

PREVIEW

List of Figures

PREVIEW

PREVIEW

PREVIEW

PREVIEW

List of Tables

PREVIEW

PREVIEW

Chapter 1.

Introduction

1.1 Background

According to the Standish group in 2009 [13], a large percentage (68%) of software projects continue to be challenged, or fail outright, in their delivery of systems costing companies $ 140 Billion. While many project management and governance methodologies have been adopted by organizations with some success, companies are still often caught in increasingly difficult situations where resources for the development of new systems, as well for the maintenance of existing infrastructure, are more and more constrained. At the same time, development of new systems can be a critical competitive factor. In order to balance these competing objectives, organizations have to be able to predict the cost and impact of projects they need to undertake. Overextending resources can lead to a risk of project failure. On the other hand, according to Shumsky [46], delaying work that could be accomplished has opportunity costs that can be nearly as damaging [1].

PREVIEW

According to the ubiquitous 2009 Standish CHAOS report [13], Software development project failure continues to be a significant risk to organizations. Ross organizes the sources of project risk into two categories: private and market risk. Private risk consists of components which affect the internal workings of a project (schedule slip, high defect rate, staff turnover etc.), Market risk consists of components which are external to a

project (cost of labor, interest rates, demand for the new product or service under development etc.). The dynamic between these two sources of risk is a crucial balance that can be the difference between a successful project and a failed project.

We present three models of software estimation for managing software development projects which represent diverse ends of the spectrum in software project management: the traditional approaches such as embodied by the PMBOK and Prince2 methodologies [42], [45], adaptive project management as proposed by Highsmith [32] and the agile approaches as summarized by Cohen and Williams [14][52]. We propose that these apparently disparate approaches are working to solve the same problem: effective software development project management. In all three cases, effective management of software development means completion of requested features in the time frame and cost estimated. We will use financial options theory technique called Real Options Theory [39] to show that these approaches may be valid under given circumstances depending on the balance between private risk and market risk (see Table 1) in the software development project environment. Market software development project risk drivers have the effect of increasing the need to speed delivery of the project and reduce costs. Private software development project risk drivers have the effect of increasing the need to focus on quality delivery (potentially at an increased cost and schedule duration).

PREVIEW

Private	**Market**
Project canceled	How much will clients be willing to pay? (market demand)

System goes sour	How much will skilled programmers cost?
Business misunderstood	How uncertain are fixed costs? Overhead?
Business changes	How well is the economy doing?
False feature rich	Where are short term interest rates heading?
Schedule slips	
Staff turnover	
Technology	

Table 1 – Ross' sources of project risk

PREVIEW

Since projects are significant investments for an organization, poor estimation in the planning of projects has a cascading effect on other projects / investments. One defensive approach project managers can take is to include some contingency in the plan and estimates. While this might work for the specific project, it has the effect of pushing out projects which could have been implemented. This opportunity cost is a hidden cost of poor estimation.

In this dissertation we will introduce granular simulation models that we will compare against a case study project executed at a Fortune 500 company in 2006. The case study project used a more traditional Unified Process software development life cycle as well as other traditional corporate governance tools for Sarbanes-Oxley / SAS 70 compliance, such as COSO-COBIT, ITIL, and the Project Management Institute's PMBOK. The specific adaptation of these standards will be detailed in several appendices.

1.2 Problem statement

Companies are more pressed than ever to use their IT resources in an effective manner to compete and survive. According to the 2009 Standish Group CHAOS report [13], large percentages (24 %) of projects taken on were canceled by the companies that undertake their implementation. An additional 44 % of projects went over budget by an average of 189% of their original estimate. Only 32 % successfully met their schedule, budget and scope objectives.

PREVIEW