

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

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.



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
Chairperson of Dissertation Committee

Date

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.

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

List of Figures	xiv
List of Tables	xviii
Chapter 1 – Introduction.....	1
1.1 Background.....	1
1.2 Problem statement.....	4
1.3 Purpose of this research	8
1.4 Project management approaches.....	14
1.5 Traditional Approaches to Project Management	17
1.6 PMI’s Project Management Body of Knowledge.....	18
1.7 Case Study Project Methodology.....	22
1.7.1 Change Definition.....	22
1.7.2 Approach.....	24
1.7.3 Change Methodology.....	24
1.7.4 Organizational taxonomy.....	27
1.7.5 Model Implementation Issues.....	29
1.7.6 Governance Plan	29
1.7.7 Communication Plan.....	30
1.7.8 Implementation Plan	31
1.7.9 Data collection	32
1.7.10 The Baseline.....	32

1.7.11 Post Implementation analysis	33
1.7.12 Case Study Project Management Approach	35
Chapter 2 – Software Cost Estimation.....	57
2.1 COCOMO	57
2.2 COCOMO II	58
2.3 SEER-SEM	61
2.4 SLIM.....	61
2.5 Other estimation models	64
Function Points	64
2.6 Projects as investments	67
2.7 Real Option Theory: application to software projects	69
2.8 Estimation of effort from Use Cases.....	74
2.8.1 Categorize actors in the use case model	78
2.8.2 Categorize the Use Cases.....	79
2.8.3 Adjust the UUCP based on technical and environmental factors	80
2.8.4 Calculate adjusted use case points (UCP).....	83
2.9 Requirements	85
2.10 Effects of requirement volatility on time and effort estimation.....	87
Chapter 3 – Research Methodology.....	91
3.1 Modeling objectives: purpose, complexity and fidelity.....	91
3.1.1 Fast and Frugal Heuristics	92
3.2 Modeling approach	94

3.3 Parametric modeling with a workflow process simulation.....	98
3.3.1 Feature requests	100
3.3.2 Feature effort required	102
3.3.3 Feature implementation cycle	103
3.3.4 Integration cycle.....	105
3.3.5 Product acceptance and rework	106
3.4 Impact of complexity on resource needs	108
3.5 Building a model.....	111
3.5.1 Capture the process	112
3.5.2 Capture the time and effort	113
3.6 Case Study Model	114
3.7 Case Study Project effort estimation approaches.....	116
Chapter 4 - Analysis.....	119
4.1 First version of the model	121
4.1.1 Changing requirements scenario.....	126
4.1.2 Resource constrained scenario.....	129
4.1.3 Resource rich scenario	132
4.2 2nd version (detailed granular) Model.....	134
4.2.1 Model Calibration	138
4.2.2 Baseline configuration	141
4.2.3 Resource optimization	143
4.2.4 Sensitivity analysis – early constraint scenario	150

4.2.5 Sensitivity Analysis - Serious delays	158
4.2.6 General observations:.....	163
Chapter 5 – Research Findings.	167
5.1 Approach 1 – Broad Parametric Modeling	167
5.2 Approach 2 – Requirements based estimation.....	169
5.3 Approach 3 – Dissertation models.....	170
5.4 Impact of estimation – an Efficient Frontier Analysis for portfolio selection and sensitivity analysis	174
Chapter 6 - Contributions and Limitations of the Research.	180
6.1 Contributions of the research	180
6.2 Limitations of the research.....	184
Chapter 7 – Future Research.....	185
Appendices.....	190
Appendix A – Auxiliary Data on Corporate failures	190
Appendix C – COCOMO sample	197
Appendix D –Agile principles	198
Appendix E – A proposed pattern language for the agile gathering of software requirements used in the case study project.....	201
Gathering software requirements.....	202
Structure Patterns	206
Content Patterns	219
Validation Patterns.....	231

Appendix F – Project Governance standards used in the case study project.....	239
Sarbanes Oxley	239
SOX Section 302.....	239
SOX Section 404a.....	240
SAS 70	240
Relevant standards	242
ITIL.....	242
ITIL Key Processes.....	242
Appendix G – Sample Root Cause Analysis (RCA) Process used in the case study project	245
Appendix H - Software Development Methodology used in the Case Study project	248
Appendix I – Software Development Model v1.0 – moderate granularity.....	282
Appendix J – Software Development Model v2.0 – Iterative process, detailed granularity	292
Appendix K: Data – Case Study – business process re-engineering project at Fortune 500.....	306
Appendix L - Case Study Project Data.....	308
Appendix M – Model data	318
References.....	329

PREVIEW

List of Figures

Figure 1 - SLIM Rayleigh distribution of effort	62
Figure 2 – parametric / process flow software estimation model	99
Figure 3 – Dissertation model flow component as a source of Feature Requests	100
Figure 4 - Dissertation model feature effort required construct	102
Figure 5 - Dissertation model feature implementation construct.....	103
Figure 6 - Dissertation model Resource level selector construct.....	104
Figure 7 - Dissertation model Integration cycle construct.....	105
Figure 8 - Dissertation model Rework construct.....	106
Figure 9 - Effort Estimating factors used in the Case Study project	118
Figure 10 - Dissertation Model Feature Request construct	121
Figure 11 – Dissertation Model Resource selection construct.....	122
Figure 12 - Sample model run - 1st version (gross granular) model	124
Figure 13 - Model example - balancing effort required vs effort available.....	125
Figure 14 - 1st version (gross granular) model, changing features requested scenario ..	126
Figure 15 – Dissertation model results, additional requirements scenario	128
Figure 16 - Dissertation 1st version (gross granular) model, resource constrained scenario	129
Figure 17 - Dissertation 1st version (gross granular) model, resource constrained scenario results	130

Figure 18 - Dissertation 1st version (gross granular) model, resource rich scenario	132
Figure 19 - Dissertation 1st version (gross granular) model, resource rich scenario results	133
Figure 20 - Dissertation 2nd version (detailed granular) Model.....	136
Figure 21 - Dissertation 2nd version (detailed granular) model baseline configuration results	141
Figure 22 - Dissertation 2nd version (detailed granular) model, resource optimization run	143
Figure 23 - Dissertation 2nd version (detailed granular) model, resource optimized scenario Analysis view	144
Figure 24 - Dissertation 2nd version (detailed granular) model, resource optimized scenario Design view	146
Figure 25 - Dissertation 2nd version (detailed granular) model, resource optimized scenario coding view.....	148
Figure 26 - Dissertation 2nd version (detailed granular) model, resource optimized scenario Test Design view	148
Figure 27 - Dissertation 2nd version (detailed granular) model, resource optimized scenario Testing and Integration view	149
Figure 28 - Dissertation 2nd version (detailed granular) model, early resource constraint scenario	150
Figure 29 - Dissertation 2nd version (detailed granular) model, early resource constraint scenario, Analysis view	151

Figure 30 - Dissertation 2nd version (detailed granular) model, early resource constraint scenario, Design view	152
Figure 31 - Dissertation 2nd version (detailed granular) model, early resource constraint scenario, coding view.....	154
Figure 32 - Dissertation 2nd version (detailed granular) model, early resource constraint scenario, Test Design view	156
Figure 33 - Dissertation 2nd version (detailed granular) model, early resource constraint scenario, testing and integration view.....	157
Figure 34 - Dissertation 2nd version (detailed granular) model, highly resource constraint scenario	158
Figure 35 - Dissertation 2nd version (detailed granular) model, highly resource constrained scenario, Analysis view.....	160
Figure 36 - Dissertation 2nd version (detailed granular) model, highly resource constrained scenario, Design view.....	160
Figure 37 - Dissertation 2nd version (detailed granular) model, highly resource constrained scenario, Coding view	161
Figure 38 - Dissertation 2nd version (detailed granular) model, highly resource constrained scenario, Test Design view.....	161
Figure 39 - Dissertation 2nd version (detailed granular) model, highly resource constrained scenario, Testing and Integration view.....	162
Figure 40 - Efficient Frontier Analysis – optimal project selection	177
Figure 41 - Efficient Frontier Analysis – based on up to 30% over estimation.....	179

Figure 42 - Sample COCOMO II calculation	197
Figure 43 - Sample Root Cause Analysis process used in Case Study project.....	247
Figure 44 - Case Study Project Methodology	274
Figure 45 - Methodology for outsourced projects	276
Figure 46 - Tailoring the methodology	279
Figure 47 - Fast tracking	280
Figure 48 - Case Study Project SDLC	281
Figure 49 - 1st version (gross granular) Model data	282
Figure 50 - 1st version (gross granular) Model	283
Figure 51 - 2nd version (detailed granular) Model	292

List of Tables

Table 1 – Ross’ sources of project risk.....	3
Table 2 – Standish-Chaos reports: 1994-2009	5
Table 3 - Project Management processes used in the Case Study project	19
Table 4 - IT Governance Methodology used in Case Study project.....	26
Table 5 - Process Implementation Domains	28
Table 6 - Project quality gate ways used in Case Study project	34
Table 7 - PMBOK Processes used in Case Study project.....	35
Table 8 - Function Point inputs.....	65
Table 9 - Use Case Point actor categories	78
Table 10 - Use Case Types	79
Table 11 - Use Case Points technical complexity factors.....	80
Table 12 - Use Case Points Environmental Factors.....	81
Table 13 - Agile Principles for gathering requirements.....	86
Table 14 - Broad Parametric Models	108
Table 15 - Use Case Technical Factors used in Case Study project estimation	116
Table 16 - Use Case Environmental Factors used in Case Study project estimation	117
Table 17 - Estimating Factors used in the Case Study project	117
Table 18 - Model Analysis Factors.....	138
Table 19 - Model Design Factors.....	139
Table 20 - Model Coding Factors	139

Table 21 - Model Test Design Factors.....	139
Table 22 - - Model Testing and Integration Factors	140
Table 23 - Predicted versus actual results.....	171
Table 24 - Sample portfolio selection using Efficient Frontier Analysis	175
Table 25 - Project alignment against strategic drivers.....	176
Table 26 – Optimal Project portfolio selection.....	176
Table 27 - Project Portfolio based on up to 30% over estimation	178
Table 28 - Corporate Software Failures.....	190
Table 29 - Key IT Governance processes used in the Case Study project	244
Table 30 - Project Inception process used in Case Study project	256
Table 31 - Project Elaboration process used in Case Study project	257
Table 32 - Project Validation process used in Case Study project	259
Table 33 - Project Implementation process used in Case Study project.....	261
Table 34 - Use Cases from the Case Study project.....	307
Table 35 - Case Study project data	308
Table 36 - Optimized Model Data – Minimum Resources Required	318
Table 37 - Optimized Model Data - resources available	321
Table 38- Model Data - Resources aligned to non-fractional units.....	323
Table 39 - Model Data - Resources required	325
Table 40 - Model Data - Resources available.....	327

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].

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).

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

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.