The Fraunhofer IESE Series on Software and Systems Engineering

Adam Trendowicz

Software Cost Estimation, Benchmarking, and Risk Assessment

The Software Decision-Makers' Guide to Predictable Software Development

Springer

CoBRA®

Adam Trendowicz
Fraunhofer IESE
Kaiserslautern
Germany

ISSN 2193-8199 ISSN 2193-8202 (electronic)
ISBN 978-3-642-30763-8 ISBN 978-3-642-30764-5 (eBook)
DOI 10.1007/978-3-642-30764-5
Springer Heidelberg New York Dordrecht London

Library of Congress Control Number: 2012948008

ACM Codes: D.2.9, K.6.1, K.6.3

Trademark Notice

CoBRA is registered trademark no. 398276811 at the German Patent and Trade Mark Office

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

About this Series

Whereas software engineering has been a growing area in the field of computer science for many years, systems engineering has its roots in traditional engineering. On the one hand, we still see many challenges in both disciplines. On the other hand, we can observe a trend to build systems that combine software, microelectronic components, and mechanical parts. The integration of information systems and embedded systems leads to so-called cyber-physical systems.

Software and systems engineering comprise many aspects and views. From a technical standpoint, they are concerned with individual techniques, methods, and tools, as well as with integrated development processes, architectural issues, quality management and improvement, and certification. In addition, they are also concerned with organizational, business, and human views. Software and systems engineering treat development activities as steps in a continuous evolution over time and space.

Software and systems are developed by humans, so the effects of applying techniques, methods, and tools cannot be determined independent of context. A thorough understanding of their effects in different organizational and technical contexts is essential if these effects are to be predictable and repeatable under varying conditions. Such process-product effects are best determined empirically. Empirical engineering develops the basic methodology for conducting empirical studies, and uses it to advance the understanding for the effects of various engineering approaches.

The series presents engineering-style methods and techniques that foster the development of systems that are reliable in every aspect. All the books in the series emphasize the quick delivery of state-of-the-art results and empirical proof from academic research to industrial practitioners and students. Their presentation style is designed to enable the reader to quickly grasp both the essentials of a methodology and how to apply it successfully.

Scientists build to learn; Engineers learn to build.

Frederick P. Brooks

Foreword by Prof. Dr. Dieter Rombach

Software Engineering is concerned with the development of large and complex software-intensive systems and services in an economical and timely manner by following engineering principles and applying best practice methods, techniques, and tools. Software is entering domains it never belonged to in the past and must face challenges it never had to confront before. High demands on software-intensive systems and increasing competitiveness within the software business have triggered a push towards systematic software engineering approaches, including techniques for managing software projects.

Applied research institutions such as the Fraunhofer Institute for Experimental Software Engineering (IESE) support software organizations in transferring into daily practice innovative, empirically proven software engineering solutions that are driven by their specific needs.

This book is the result of a successful collaboration between the process management division of Fraunhofer IESE and many software companies in the field of software engineering technology transfer. The book introduces an innovative software management technology called CoBRA, which has been deployed in a number of software companies.

Fraunhofer IESE developed the Cost Estimation, Benchmarking, and Risk Assessment method (CoBRA) driven by industrial needs with respect to managing software project resources. In addressing project management objectives, CoBRA goes far beyond simply predicting the development effort. It supports project decision-makers in negotiating the project scope, managing project risks, benchmarking productivity, and directing improvement activities. At the same time, it meets typical constraints encountered in software engineering contexts where other estimation methods typically fail. The method requires neither large amounts of project measurement data nor extensive involvement of human experts. While many leading software engineering researchers and practitioners agree on the need for a systematic approach for combining quantitative data with human judgment, CoBRA is actually doing it.

What makes this book special is that it is driven by industrial practice and aimed at industrial practice. The book introduces the principles of the CoBRA method followed by the basic procedures for applying the method. The methodological concepts are illustrated by a number of practical examples, and the use of the method is exemplified by several case studies from various software organizations.

For any software organization that does not want to leave the success of its software projects to chance, this book should serve as a standard handbook.

Kaiserslautern, Germany

Prof. Dr. Hans Dieter Rombach
Executive Director
Fraunhofer Institute for Experimental Software Engineering

Foreword by Dr. Koichi Matsuda

Recently, we experienced the so-called 2007 Problem in Japan. Within a short period of time, the Japanese industry, including the software companies, lost a complete generation of highly skilled senior professionals. Born during the time of the baby boom, they now turned 60 and left their jobs to retire. The Software Engineering Center (SEC) was asked to undertake activities to prevent a negative impact of the "2007 Problem" on the software industry. One such undertaking was skill and technology transfer between age groups. Still, in many software engineering areas, professional expertise is rather hard to grasp and to share. Software project effort estimation is one such area. The success of estimation is heavily dependent on human expertise, which is hidden in the experts' minds and is difficult to grasp. This book is unique in that it provides a practical solution to this problem.

Today, software processes benefit greatly from advanced methodological and tool support. Many process areas that used to rely heavily on human knowledge and skills are now largely independent of it. However, the area of project estimation is still largely dependent on experienced engineers. The CoBRA method presented in this book is unique in that it systematically acquires expertise hidden in the minds of human experts and transforms it into explicit knowledge that is easier to share between people and projects.

In Japan, the term KKD is often used in the context of software development. The first "K" stands for *Keiken* (*experience*), the second "K" stands for *Kan* (*intuition*), and "D" stands for *Dokyo* (*courage*). CoBRA improves the *courage* element by combining quantitative project data with the *experience* and *intuition* of human experts. In CoBRA, project estimation *intuition* is enhanced by a systematic, yet comprehensible, methodology for acquiring and documenting the qualitative knowledge of experienced engineers and by integrating it with quantitative data. In this sense, CoBRA introduces a scientific basis to the KKD paradigm.

I am convinced that the software industry, not only in Japan, can greatly benefit from using the CoBRA method for managing the greatest organizational asset, which is knowledge. This book explains the CoBRA method and shows how to use it to achieve and maintain software project excellence.

Tokyo, Japan

Dr. Koichi Matsuda
Director, Software Engineering Center
Information-technology Promotion Agency

Quotes from Industry

The CoBRA method is a way of integrating the “Art” and the “Science” of software estimation, which is usually believed to be a “Black Art.” The CoBRA method demystifies the Black Art of software estimation.

–Yasushi Ishigai
Research Director at Research Center for Information Technology
Mitsubishi Research Institute, Inc., Japan

We had not been able to imagine building our own software estimation models until we encountered the CoBRA method. As far as combining project data and expert judgment for the purpose of software effort estimation is concerned, we can definitely say that there are no other methods that are comparable to CoBRA.

–Morihiko Shinoda, Deputy Department Manager
–Yutaka Masaoka, Senior Engineer
Government, Public Sector Systems Division
Hitachi Solutions, Ltd, Japan

We used the CoBRA method for early-stage estimation of system integration projects. We were very satisfied with CoBRA because we could easily model relevant cost drivers that are specific to our own context as well as base development productivity. From limited use in one group, we have now expanded its use to department-wide activities.

–Yasushi Aizaki
Manager at Systems Development Division
NTT Data Sekisui Systems, Japan

I am convinced that the CoBRA method has a high potential of resolving “acquisition issues” related to the accountability of software costs in IT business, which enduringly persist on the side of IT customers. That’s because the models are very simple and easy to build and can be understood even by non-IT professionals.

–Hiroshi Iwakiri
General Manager
Information Systems Business Unit
Mitsubishi Electric Corporation

Preface

What This Book Is About?

In this book, we present a method for estimating the effort required to successfully complete a software development project. The method is called Cost Estimation, Benchmarking, and Risk Assessment—CoBRA for short—and combines human judgment and measurement data in order to systematically create a custom-specific effort estimation model.

The book provides a comprehensive specification of processes for developing the CoBRA effort model and for applying the model in a number of different project management scenarios. For each of these processes, we describe detailed activities that need to be performed as well as associated techniques. We illustrate the presented concepts with a number of examples and graphical illustrations. Moreover, we provide a series of practical guidelines on how to apply these processes, based on industrial experiences regarding project effort estimation in general, and on using the CoBRA method in particular.

Furthermore the book reports several real-world cases where the CoBRA method was applied in various industrial contexts in order to illustrate the practical usage of the method. The cases represent different estimation contexts in terms of software project environment, estimation objectives, and estimation constraints.

Objectives of This Book

The main objective of this book is to present the Cost Estimation, Benchmarking, and Risk Assessment Method (CoBRA) in a way that allows for applying it successfully in practical situations. Consequently, the key goals we are aiming at with this book include:

- Complete and comprehensible specification of all relevant CoBRA processes such as developing and applying the effort estimation model. This includes the

description of process activities, their inputs and outputs, the personnel involved, and the theories and techniques employed.
- Comprehensive explanation of the presented concepts through practical examples, graphical illustrations, and guidelines from practice.
- Illustration of real-world CoBRA usage through exemplary application cases from various industrial contexts.

After reading this book a reader should understand the principles of the CoBRA method, know the basic CoBRA processes, and be able to adapt and use the method in a specific context.

To Whom This Book Is Addressed

Software Practitioners

We addressed this book to all software practitioners who deal with planning and managing software development projects as part of their daily work. This group includes primarily—but is not limited to—project managers and project estimators. In order to facilitate understanding and practical application of the concepts presented in the book, we illustrate them with a number of practical examples and guidelines. In particular, the book is addressed to those software practitioners who would need an alternative to expensive estimation based on expert judgment, yet who do not have sufficient measurement data to employ analytical effort estimation.

Students

The book is also addressed to students of software engineering and of associated courses. In order to support in-depth study of the concepts presented in the book, we include descriptions of associated theoretical foundations and refer to appropriate further readings.

Key Terminology Used in This Book

In this book, we use several basic terms that in other literature and in practice are often used interchangeably, although they do have different meanings. In order not to confuse the reader, we would like to start by clarifying the most important terms we will use throughout the text. For a more comprehensive dictionary of the employed terms, please refer to the Glossary at the end of the book.

Cost Versus Effort

Although principally and intuitively different, the terms "cost" and "effort" are usually used as synonyms in the software project management area. Webster's dictionary defines cost as "the amount or equivalent paid or charged for something" and effort as "conscious exertion of power" or "the total work done to achieve a particular end." In the software engineering domain, cost is defined in a monetary sense, and with respect to a software development project, it refers to the partial or total monetary cost of providing (creating) a certain product or service. Effort, on the other hand, refers to manpower spent on performing activities aimed at providing a certain product or service. In consequence, project cost includes, but is not limited to, project effort. In practice, cost includes such elements as fixed infrastructure and administrative costs. Moreover, depending on the project context (e.g., currency or cost of manpower unit), project costs may differ even if the project effort is the same.

In software engineering literature and practice, "cost" is often used as a synonym for "effort." One way to notice the difference is to look at units used. Cost in a monetary sense is typically measured in terms of certain currencies (e.g., $, €, ¥, etc.), whereas cost in an effort sense is typically measured as manpower (e.g., person-hours, person-days, person-months, etc.).

In this book, we focus on estimating software development effort, and we consistently differentiate between cost and effort.

Estimation Versus Prediction Versus Planning

In software engineering, effort estimation, prediction, and planning are related to each other; yet, they have different meanings, i.e., they refer to different project management activities. Actually, the dictionary definitions perfectly reflect the differences between these three processes:

- *Estimation*: "the act of judging tentatively or approximately the value, worth, or significance of something"
- *Prediction*: "the act of declaring or indicating in advance; especially foretelling on the basis of observation, experience, or scientific reason"
- *Planning*: "the act or process of making or carrying out plans; *specifically* the establishment of goals, policies, and procedures for a social or economic unit"

Adam Trendowicz

Acknowledgments

This book is not the fruit of one man's work—although it has one author. A number of great people have made their explicit or implicit contribution to this book. Here I would like to thank them for their efforts.

First of all I would like to thank Prof. Dr. Dieter Rombach for creating the Fraunhofer Institute for Experimental Software Engineering—a great environment for innovative work.

I would like to express thanks to Mr. Yasushi Ishigai for his great support in deploying the CoBRA method in Japan and for many inspiring discussions regarding the method. I would also like to thank Mr. Ishigai for his great dedication in reviewing the initial version of this book and making superb comments, which helped me to greatly improve the book.

Furthermore, I would like to thank the representatives of various software companies where the CoBRA method was applied. They provided great support in transferring the CoBRA method into industrial practice and provided valuable feedback about the method from a practical perspective. In particular, I would like to thank Mr. Hisato Hatano, Dr. Nahomi Kikuchi, Mrs. Haruka Nakao, and Mr. Debanjan Ray.

I would like to express many thanks to the reviewers of the book, Mr. Michael Kläs and Mrs. Sylwia Kopczyńska, for their valuable remarks.

Last but not least I would like to thank Mrs. Sonnhild Namingha, the very first reader of the complete book, who revised its spelling and copyedited it.

Contents

Part I Predictable Software Development

Part II The CoBRA Method

Part III Industrial Applications

Acronyms

Allette	Allette Systems Pty. Ltd (company name)
ARE	Acceptable Risk Exposure
ARL	Acceptable Risk Level
CMMI	Capability Maturity Model Integrated
CoBRA	Cost Estimation, Benchmarking, and Risk Assessment
DF	Direct effort Factor
EF	Experience Factory
EMB	Embedded Software Systems
EO	Effort Overhead
Ext	Extreme
FhG/IESE	Fraunhofer Institute for Experimental Software Engineering
GUI	Graphical User Interface
IF	Indirect effort Factor
IPA/SEC	Information-technology Promotion Agency, Software Engineering Center
ISO	International Standardization Organization
IV&V	Independent Verification and Validation
JAMSS	Japan Manned Software Systems (company name)
LHRO	Performance-Optimized Latin Hypercube (simulation technique)
Max	Maximal
MC	Monte Carlo (simulation technique)
MCDA	Multiple Criteria Decision Analysis
Min	Minimal
MIS	Management and Information Systems
ML	Most Likely
MMRE	Mean Magnitude of Relative Error (or Mean Magnitude of Relative Estimation Error)
MRE	Magnitude of Relative Error (or Magnitude of Relative Estimation Error)
Nom	Nominal
Oki	Oki Electric Industry, Ltd (company name)
PERT	Program Evaluation and Review Technique
PMBOK	Project Management Body of Knowledge

PMI	Project Management Institute
PMO	Project Management Office
RE	Relative Error (or Relative Estimation Error)
sd&m	software design & management AG (company name; currently Capgemini Deutschland Holding GmbH)
SEPG	Software Engineering Process Group
SISL	Siemens Information Systems, Ltd (company name)