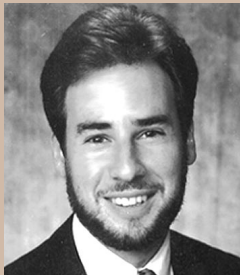


Sitting on the Suitcase

Suppose you're preparing for a trip and deciding which suitcase to take. You have a small suitcase that you like because it's easy to carry and will fit into an airplane's overhead storage bin. You also have a large suitcase, which you do not prefer because you'll have to check it in, lengthening your trip. You lay your clothes beside the suitcase, and it appears that they'll almost fit into the small suitcase. What do you do? You might try packing them very carefully, not wasting any space, and hoping they all fit. If that approach doesn't work, you might try stuffing them into the suitcase with brute force, sitting on the top and trying to squeeze the latches closed. If that still doesn't work, you're faced with a difficult choice:



leave a few clothes at home or take the larger suitcase.

Software projects face a similar dilemma. Project planners often find a gap between a project's business targets and its estimated schedule and cost. If the gap is small, the planner might be able to control the project to a successful conclusion by preparing extra carefully or by squeezing the project's schedule, budget, or feature set. If the gap is large, the project's targets must be reconsidered.

Software Estimation

Some industry experts imply that the goal of estimation is to achieve pinpoint accuracy. They claim that effort estimates created using automated estimation tools can

be accurate to within about 10%.¹ Meanwhile, various reports about software industry practices suggest that real-world estimation accuracy falls far short of this ideal. Effort estimates that are accurate to within 50% are found in less than half of all projects.²

Estimation accuracy is probably worse than it at first appears. The Standish Group's 1994 "Chaos Report" found that "challenged projects" (those that experienced schedule and budget overruns) routinely threw out significant amounts of functionality in order to deliver the schedules and budgets they eventually did. Of course, their estimates weren't for the abbreviated versions they eventually delivered; they were for the originally specified, full-featured versions. If these late projects had delivered all of their originally specified functionality, they would have overrun their plans even more.

These cost and schedule overruns are partly attributable to software developer optimism.³ They are partly attributable to the use of inefficient practices that fall short of expectations. And they are partly attributable to unrealistic target setting—targets are established, the targets become commitments, and the commitments are later reported as "estimates."

Target Setting and Control

The purpose of a software estimate is not to make a perfect prediction about a project's eventual cost or schedule. Changes occur throughout a project that invalidate many of the assumptions that went into

EDITOR-IN-CHIEF:

Steve McConnell
10662 Los Vaqueros Circle
Los Alamitos, CA 90720-1314
software@construx.com

EDITORS-IN-CHIEF EMERITUS:

Carl Chang, Univ. of Illinois, Chicago
Alan M. Davis, Omni-Vista

EDITORIAL BOARD

Maarten Boasson, Hollandse Signaalapparaten
Terry Bollinger, The MITRE Corp.
Andy Bytheway, Univ. of the Western Cape
David Card, Software Productivity Consortium
Larry Constantine, Constantine & Lockwood
Christof Ebert, Alcatel Telecom
Martin Fowler, ThoughtWorks
Robert L. Glass, Computing Trends
Lawrence D. Graham, Black, Lowe, and Graham
Natalia Juristo, Universidad Politécnica de Madrid
Warren Keuffel
Karen Mackey, Cisco Systems
Brian Lawrence, Coyote Valley Software
Tomoo Matsubara, Matsubara Consulting
Stephen Mellor, Project Technology
Wolfgang Strigel, Software Productivity Centre
Jeffrey M. Voas, Reliable Software Technologies Corp.
Karl E. Wieggers, Process Impact

INDUSTRY ADVISORY BOARD

Robert Cochran, Catalyst Software
Annie Kuntzmann-Combelles, Objectif Technologie
Enrique Draier, Netsystem SA
Eric Horvitz, Microsoft
David Hsiao, Cisco Systems
Takaya Ishida, Mitsubishi Electric Corp.
Dehua Ju, ASTI Shanghai
Donna Kasperson, Science Applications International
Günter Koch, Austrian Research Centers
Wojtek Kozaczynski, Rational Software Corp.
Masao Matsumoto, Univ. of Tsukuba
Dorothy McKinney, Lockheed Martin Space Systems
Susan Mickel, BoldFish
Deependra Moitra, Lucent Technologies, India
Dave Moore, Vulcan Northwest
Melissa Murphy, Sandia National Lab
Kiyoh Nakamura, Fujitsu
Grant Rule, Guild of Independent Function Point Analysts
Girish V. Seshagiri, Advanced Information Services
Chandra Shekaran, Microsoft
Martyn Thomas, Praxis
Rob Thomsett, The Thomsett Company
John Vu, The Boeing Company
Simon Wright, Integrated Chipware
Tsuneo Yamaura, Hitachi Software Engineering

MAGAZINE OPERATIONS COMMITTEE

Sorel Reisman (chair), William Everett (vice chair),
James H. Aylor, Jean Bacon, Thomas J. (Tim)
Bergin, Wushow Chou, George V. Cybenko,
William I. Grosky, Steve McConnell, Daniel E.
O'Leary, Ken Sakamura, Munindar P. Singh, James
J. Thomas, Yervant Zorian

PUBLICATIONS BOARD

Sallie Sheppard (vice president), Sorel Reisman
(MOC chair), Rangachar Kasturi (TOC chair), Jon
Butler (POC chair), Angela Burgess (publisher),
Laurel Kaleda (IEEE representative), Jake Aggarwal,
Laxmi Bhuyan, Lori Clarke, Alberto del Bimbo, Mike
Liu, Mike Williams (secretary), Zhiwei Xu

FROM THE EDITOR

early estimates, and even if we eventually achieve results within 10% of our original assessment, the results are usually a fluke. The project we end up with is rarely the same project we originally estimated, so the

“accurate” estimate was really for some other project.

I propose a different goal for software estimation: An estimate should achieve accuracy sufficient to let the project manager control the project

Call for Articles & Reviewers

Global Software Development

Publication: January/February 2001

Submission deadline: 30 June 2000

Economic forces are relentlessly turning national markets into global markets and turning product development into a multi-site, multicultural, geographically distributed undertaking. Global Software Development is the new business paradigm that organizations are adopting to meet today's challenges and derive competitive advantage. Articles should be experience-based and useful to the software community at large. Topics of interest include

- configuration management and builds
- around-the-clock development, concurrent software engineering, and multiple sites
- experiences with collaboration technologies
- techniques for distributing work across sites to minimize dependencies
- knowledge transfer, knowledge management, and informal sharing
- overcoming cultural and language barriers

Guest Editors

James D. Herbsleb, Bell Laboratories
herbsleb@research.bell-labs.com
Deependra Moitra, Lucent Technologies
dmoitra@acm.org

For detailed author guidelines, visit computer.org/software or e-mail us at software@computer.org.

to meet its business targets. As Tom Gilb points out, estimates by themselves merely give us the ability to predict project outcomes, but we don't need just prediction; we need control.⁴


Estimation on software projects interplays with both target setting and control. Businesses have many important reasons to establish targets independent of software estimates, just as you might have important reasons to prefer a small suitcase to a large one. Sometimes software must be ready for a trade show, or for a holiday sales season, or for tax preparation season, or for some other externally imposed date. Sometimes a business is operating under tight cost constraints and will be penalized for exceeding cost targets. The business environment dictates these targets, and a company has little ability to influence them. What a business can influence are the parameters of its software projects. In this context, a pinpoint-accurate estimate that "we will deliver our software four weeks after the end of the holiday sales season" is of little use. The value to the business arises from being able to control the project enough to deliver desirable functionality within the business timeframe and cost desired.

Thus, the primary purpose of software estimation is not to predict a project's outcome; it is to determine whether a project's targets are realistic enough to allow the project to be controlled to meet its targets. Will the clothes I want to take on my trip fit into the small suitcase or will I be forced to take the large one? Can I take the small suitcase if I make minor adjustments? Business managers want the same kinds of answers. They often don't want an accurate estimate that tells them that the desired clothes won't fit into the suitcase; they want a plan for making as many of the clothes fit as possible.

Excess Baggage

Problems arise when the gap between the business targets and the schedule and effort needed to achieve

those targets becomes too large. I assert (with no real proof other than my personal experience) that if the initial target and initial estimate are within about 20% of each other, the project manager will have enough maneuvering room to control the feature set, schedule, team size, and other parameters to meet the project's business goals. If the gap between the target and what is actually needed is too wide, the manager will not be able to control the project to a successful conclusion by making minor adjustments to project parameters. No amount of careful packing or sitting on the suitcase will squeeze all my clothes into that smaller suitcase, and I'll have to take the larger one even if it isn't my first choice. The project targets will need to be brought into better alignment with reality before the manager can control the project to meet its targets.

Estimates don't need to be perfectly accurate as much as they need to be useful. We should not let the pursuit of ever more accurate estimates blind us to the important but limited role that estimation plays. Control is the all-important suitcase. The business targets are the suitcase's valuable contents. Accurate estimates just provide the wheels that make the suitcase a little bit easier to maneuver. 

References

1. C. Jones, *Assessment and Control of Software Risks*, Prentice Hall/Yourdon Press, Englewood Cliffs, N.J., 1994.
2. "Charting the Seas of Information Technology," The Standish Group, West Yarmouth, Mass., 1994.
3. M. van Genuchten, "Why Is Software Late? An Empirical Study of Reasons for Delay in Software Development," *IEEE Trans. Software Engineering*, Vol. 17, No. 6, June 1991, pp. 582-590.
4. T. Gilb, *Principles of Software Engineering Management*, Addison-Wesley, Reading, Mass., 1988.

DEPARTMENT EDITORS

Bookshelf: Warren Keuffel, wkeuffel@computer.org
 Culture at Work: Karen Mackey, Cisco Systems, kmackey@best.com
 Loyal Opposition: Robert Glass, Computing Trends, rglass@indiana.edu
 Manager: Don Reifer, dreifer@sprintmail.com
 Quality Time: Jeffrey Voas, Reliable Software Technologies Corp., jmvoas@rstcorp.com
 Soapbox: Tomoo Matsubara, Matsubara Consulting, matsuo@computer.org
 Softlaw: Larry Graham, Black, Lowe, and Graham, graham@blacklaw.com

STAFF

Group Managing Editor
Dick Price
 Managing Editor
Dale C. Strok
 dstrok@computer.org
 Associate Editor
Dennis Taylor
 News Editor
Crystal Chweh
 Staff Editor
Jenny Ferrero
 Assistant Editors
Cheryl Balties and Shani Murray
 Magazine Assistants
Dawn Craig and Angela Delgado
 software@computer.org
 Art Director
Toni Van Buskirk
 Cover Illustration
Dirk Hagner
 Technical Illustrator
Alex Torres
 Production Artist
Carmen Flores-Garvey
 Acting Executive Director
Anne Marie Kelly
 Publisher
Angela Burgess
 Membership/Circulation
 Marketing Manager
Georgann Carter
 Advertising Assistant
Debbie Sims

CONTRIBUTING EDITORS

Nancy Mead, Ware Myers, Keri Schreiner, Pradip Srimani, Robin Yeager

Editorial: All submissions are subject to editing for clarity, style, and space. Unless otherwise stated, bylined articles and departments, as well as product and service descriptions, reflect the author's or firm's opinion. Inclusion in *IEEE Software* does not necessarily constitute endorsement by the IEEE or the IEEE Computer Society.

To Submit: Send 2 electronic versions (1 word-processed and 1 postscript or PDF) of articles to Magazine Assistant, *IEEE Software*, 10662 Los Vaqueros Circle, PO Box 3014, Los Alamitos, CA 90720-1314; software@computer.org. Articles must be original and not exceed 5,400 words including figures and tables, which count for 200 words each.

IEEE Software