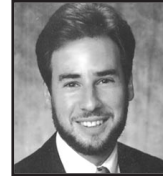




From the Editor



Brooks' Law Repealed

EDITOR-IN-CHIEF: Steve McConnell • Construx Software • software@construx.com

For more than 20 years, industry experts have been reciting Brooks' Law as gospel: Adding people to a late software project is like pouring gasoline on a fire—it just makes it later. Twenty years after the initial publication of *The Mythical Man-Month*, Fred Brooks reiterated that Brooks' Law was still “the best zeroth order approximation to the truth.”¹

In spite of Brooks' Law, adding people to a late project remains commonplace. A study of software projects in the United Kingdom found that about half of the managers of runaway projects (projects that exceeded their planned schedules or budgets by more than 30%) attempted to bring their projects under control by adding staff.² When junior project managers try to rescue a late project by adding more staff, wizened project managers and consultants intone the mantra: “Mustn't add people to a late project; adding people will make the project later.” I have evangelized this well-worn software engineering chestnut many times myself, but I no longer think it's true.

Brooks' Law is based on the idea that communications overhead is a significant factor on software projects, and that work on a software project is not easily partitioned into isolated, independent tasks. Ten people can pick cotton ten times as fast as one person because the work is almost perfectly partitionable, requiring little communication or coordination. But nine women can't have a baby any faster than one woman can because the work is not partitionable. Brooks argues that work on a software project is more like having a baby than picking cotton. When new staff are brought into a late project, they aren't immediately productive, and they must be trained. The staff who must train them are already productive, but they lose productivity while they're training new staff. Brooks argues that, on balance, more effort is lost to training and additional coordination and communications overhead than is gained when the new staff eventually becomes productive.

To those of us who have been around the software-project block a few times, the claim feels true. We've

participated on projects in which new people are brought on at the end. We know the irritation of having to answer questions from new staff when we're already feeling overwhelmed about our own work. We've seen new hires make mistakes that set the whole project back. And we've experienced additional schedule slips even after staff has been added to a late project. A typical scenario looks like this:

- ◆ Project start: A schedule of 12 months is established for the whole project.
 - ◆ Month 9: The project manager proposes adding two people, but the developers say that they're 75% done with the project and adding new staff in the final stages of the project will make the project later.
 - ◆ Month 11: The project team acknowledges that it isn't going to make its 12-month release date. After a heated debate about Brooks' Law, the project manager adds two people and reschedules the project for completion at the 14-month mark.
 - ◆ Month 14: The project isn't ready to be released. Developers chide the project manager for violating Brooks' Law.
 - ◆ Month 16: The project is finally released with significantly less functionality than originally planned.
- Did Brooks' Law contribute to this project's schedule problems? Resolution of the issue turns on project estimation, project tracking, and training.

ESTIMATION

The Standish Group's Chaos report found that the average business systems project overran its original schedule by 120%.³ Estimation accuracy on a typical software project is terrible. And it's actually worse than this statistic suggests because more than half of the late projects trimmed significant functionality in order to meet their 120-percent-behind-schedule result. Those projects would have been even later if they had delivered all of their originally planned and estimated functionality.



At Month 11, without effective project estimation in place, the plan to release the software at the 14-month mark is an exercise in ungrounded fantasy. When the software is finally released at Month 16, everyone knows that the project was completed later than its revised target of 14 months, but what no one can know with any certainty is what would have happened if new hires hadn't been added to the project. The claim is that adding staff to a late project makes it later—but later than what? Later than a systematic, well-founded estimate, or later than an estimate that was optimistic by more than 100% in the first place?

PROJECT TRACKING

The near-universality of poor project tracking (knowing a project's status with some degree of clarity) also undermines Brooks' Law. If we begin a 12-month project with five people instead of three, we incur some per-person reduction in productivity, but total productivity will be higher. What if we add two people to a 12-month project at the end of Month 2? That's still early in the project and Brooks' Law doesn't apply. Organizations such as NASA's Software Engineering Laboratory recommend starting software projects with a small senior staff and adding staff after initial requirements are done and architecture work is mostly complete.⁴ Obviously, it's beneficial to add staff until some point in the project's schedule, after which adding staff becomes detrimental. Implicit in Brooks' Law is that it applies only to the final phases of a project. The question is, How do you know whether you're in a project's final phases?

We've all participated in projects in which we arrived at the planned release date, took a three-week schedule slip, then continued slipping for six months or more after that. At the time of the first

schedule slip, we don't want to add new staff to the "late project" because three weeks wouldn't be enough time for them to become productive. But six months later we wish that we had added the new staff, because six months would have been more than enough time for them to become productive.

At Month 9 in the scenario above, without effective project tracking in place, the project manager can't know with any clarity whether the project is 75% complete or 25% complete. Studies by the Software Engineering Institute have found that poor project tracking is nearly universal in chaotic projects (which are the vast majority).⁵ Most projects think they're 90% complete for the last 50% of the project, which obviously can't be the case. Because of widespread poor tracking, people think they're at risk from Brooks' Law for significantly longer periods than they really are.

TRAINING

For Brooks' Law to be true, the amount of training effort required from existing staff must be significant. The amount of effort lost to training must exceed the productivity contributed by new staff when they eventually become productive. In Brooks' example, three people work for two months, then two more people are brought into the project, requiring one month of training each. This is absurdly conservative. How could the first three people possibly have done enough work in just two months to require a whole month of training for new staff members? The time that existing staff spends with new staff certainly does take productive time away from the project, and time spent training can be frustrating to existing staff members who are already feeling pressured to complete their own work. But the loss is nothing like the ratio needed for Brooks' Law to be true.

Programming Advice from the Experts

The Pearls are Back

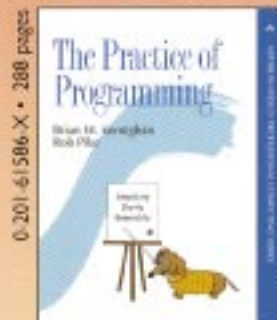


0-201-45786-0 • 256 pages • ACM Press

NEW!

Filled with lucid and witty descriptions of practical programming techniques and fundamental design principles.

Also Available



0-201-61506-X • 288 pages

Best-selling authors Brian Kernighan and Rob Pike draw on years of experience to help programmers be more effective and productive.

Check us out!
www.aw.com/cseng/

 **Addison-Wesley**



From the Editor

ZEROth-ORDER APPROXIMATION OR JUST ZERO?

"Late" chaotic projects are likely to be much later than the project manager thinks—project completion isn't three weeks away, it's six months away. Go ahead and add staff. You'll have time for them to become productive. Your project will still be later than your plan, but that's not a result of Brooks' Law. It's a result of underestimating the project in the first place. The additional people will help, not hurt.

Controlled projects are less susceptible to Brooks' Law than chaotic projects. Their better tracking allows them to know when they can safely add staff and when they can't. Their better documentation and better designs make tasks more partitionable and training less labor intensive. They can add staff later in the project with less risk to the project.

Is Brooks' Law the best zeroth-order approxima-

tion to the truth? I don't think so. Every project eventually reaches a point at which adding staff is counterproductive, but that point occurs later than Brooks' law states and in limited circumstances that are easily identified and avoided. ♦

REFERENCES

1. F.P. Brooks, Jr., *The Mythical Man-Month*, anniversary ed., Addison-Wesley, Reading, Mass., 1995.
2. A. Cole, "Runaway Projects—Cause and Effects," *Software World*, Vol. 26, No. 3, 1995, pp. 3-5.
3. The Standish Group, *Charting the Seas of Information Technology*, The Standish Group, Dennis, Mass., 1994.
4. *Recommended Approach to Software Development, Rev. 3*, Tech. Report SEL-81-305, NASA Goddard Space Flight Center, Greenbelt, Md., 1992.
5. Software Eng. Inst., *Process Maturity Profile of the Software Community—1998 Year End Update*, Software Eng. Inst., Pittsburgh, Mar. 1999.

IEEE SOFTWARE

EDITORIAL BOARD

Maarten Boasson (Hollandse Signaalapparaten), Terry Bollinger (MITRE), Andy Bytheway (Univ. of the Western Cape), David Card (Software Productivity Consortium), Carl Chang (Univ. of Ill., Chicago), Larry Constantine (Constantine & Lockwood), Christof Ebert (Alcatel Telecom), Robert Glass (Computing Trends), Lawrence D. Graham (Christensen, O'Connor, Johnson, & Kindness), Natalia Juristo (Universidad Politécnica de Madrid), Warren Keuffel, Karen Mackey (Cisco Systems), Brian Lawrence (Coyote Valley Software), Tomoo Matsubara (Matsubara Consulting), Nancy Mead (Software Eng. Inst.), Stephen Mellor (Project Technology), Pradip Srimani (Colorado State Univ.), Wolfgang Strigel (Software Productivity Centre), Jeffrey M. Voas (Reliable Software Technologies Corporation), Karl E. Wiegiers (Process Impact)

INDUSTRY ADVISORY BOARD

Robert Cochran (Catalyst Software), Annie Kuntzmann-Combelles (Objectif Technologie), Alan Davis (Omni-Vista), Enrique Draier (Netsystem SA), Eric Horvitz (Microsoft), Dehua Ju (ASTI Shanghai), Donna Kasperson (Science Applications Int'l), Günter Koch (Austrian Research Centers), Wojtek Kozaczynski (Rational Software Corp.), Masao Matsumoto (Univ. of Tsukuba), Susan Mickel (Rational Univ.), Deependra Moitra (Lucent Technologies, India), Melissa Murphy (Sandia), Kiyoh Nakamura (Fujitsu), Grant Rule (Guild of Independent Function Point Analysts), Chandra Shekaran (Microsoft), Martyn Thomas (Praxis), Sadakazu Watanabe (Fukui Univ.)

CONTRIBUTING EDITORS

Dale Adams, David Clark, Greg Goth, Ware Myers, Roger Pressman, Keri Schreiner, Ellen Ullman

MAGAZINE OPERATIONS COMMITTEE

Gul Agha (chair), James Aylor, Jean Bacon, Wushow Chou, George Cybenko, William Grosky, Steve McConnell, Daniel E. O'Leary, Ken Sakamura, Munindar P. Singh, James J. Thomas, Michael R. Williams, Yervant Zorian

PUBLICATIONS BOARD

Benjamin Wah (chair), Jake Aggarwal, Jon Butler, Alberto del Binbo, Ming T. Liu, Nancy Mead, Joseph E. Urban, Zhiwei Xu

EDITOR-IN-CHIEF: **STEVE MCCONNELL**
10662 LOS VAQUEROS CIRCLE
LOS ALAMITOS, CA 90720-1314
software@construx.com

EDITORS-IN-CHIEF EMERITUS:
CARL CHANG AND **ALAN M. DAVIS**

MANAGING EDITOR: **DALE C. STROK**
dstrok@computer.org
GROUP MANAGING EDITOR: **DICK PRICE**
ASSOCIATE EDITOR: **DENNIS TAYLOR**
NEWS EDITOR: **CRYSTAL CHWEH**
ASSISTANT EDITORS: **CHERYL BALTES,**
SHANI MURRAY, AND **JENNY FERRERO**

MAGAZINE ASSISTANT: **ROBIN MARTIN**
software@computer.org

ART DIRECTOR: **TONI VAN BUSKIRK**
COVER ILLUSTRATION: **DIRK HAGNER**
TECHNICAL ILLUSTRATOR: **ALEX TORRES**
PRODUCTION ARTIST: **LARRY BAUER**

EXECUTIVE DIRECTOR AND CHIEF EXECUTIVE OFFICER: **T. MICHAEL ELLIOT**
MEMBERSHIP/CIRCULATION
MARKETING MANAGER: **GEORGANN CARTER**
ADVERTISING MANAGER: **PATRICIA GARVEY**
ADVERTISING COORDINATOR: **DEBBIE SIMS**

Editorial: Send 2 electronic versions (1 word-processed and 1 postscript or PDF) of articles to Managing Editor, *IEEE Software*, 10662 Los Vaqueros Cir., 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. 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.

Copyright and reprint permission: Copyright © 1999 by the Institute of Electrical and Electronics Engineers, Inc. All rights reserved. Abstracting is permitted with credit to the source. Libraries are permitted to photocopy beyond the limits of US copyright law for private use of patrons those post-1977 articles that carry a code at the bottom of the first page, provided the per-copy fee indicated in the code is paid through the Copyright Clearance Center, 222 Rosewood Dr., Danvers, MA 01923. For copying, reprint, or republication permission, write to Copyright and Permissions Dept., IEEE Publications Admin., 445 Hoes Ln., Piscataway, NJ 08855-1331.

Circulation: *IEEE Software* (ISSN 0740-7459) is published bimonthly by the IEEE Computer Society. IEEE headquarters: Three Park Ave., 17th Floor, New York, NY 10016-5997. IEEE Computer Society Publications Office: 10662 Los Vaqueros Cir., PO Box 3014, Los Alamitos, CA 90720-1314; (714) 821-8380; fax (714) 821-4010. IEEE Computer Society headquarters: 1730 Massachusetts Ave. NW, Washington, DC 20036-1903. Annual electronic/paper/combo subscription rates for 1999: \$27/34/44 in addition to any IEEE Computer Society dues, \$49 in addition to any IEEE dues; \$93 for members of other technical organizations. Nonmember subscription rates available on request. Back issues: \$10 for members, \$20 for nonmembers. This magazine is available on microfiche.

Postmaster: Send undelivered copies and address changes to Circulation Dept., *IEEE Software*, PO Box 3014, Los Alamitos, CA 90720-1314. Periodicals Postage Paid at New York, NY, and at additional mailing offices. Canadian GST #125634188. Canada Post Publications Mail Product (Canadian Distribution) Sales Agreement Number 0487805. Printed in the USA.