Open Books

The Mythical Man-Month: Essays on Software Engineering (Anniversary Edition)

by Frederick P. Brooks, Jr.
Addison-Wesley
322 pages, $24.69
ISBN# 0-201-83595-9

Anybody who has hung around the technical end of large software development projects has at least heard of *The Mythical Man-Month* by Fred Brooks. This account of large-scale software development is one of the classic texts in the field of system software development.

Brooks, now a professor on the forefront of virtual reality research, was a principal manager of the IBM OS/360 project, one of the largest system software projects of the 1960s. The book, originally published in 1975, combined his firsthand experience at managing such a large project and then-current research on software engineering management. Addison-Wesley has taken advantage of the book's twentieth anniversary to publish a new edition with four additional essays.

Many maxims that are common wisdom in software engineering management today have their origins in this book: Adding additional manpower to a late software project only makes it later ("Brooks' Law"); software development schedules are invariably optimistic; a great programmer can be 10 times more productive than an average one; designs must be complete before implementation begins. On each of these points, Brooks draws extensively from published research and his own experience. On the design issue, for example, he writes, "It is a very humbling experience to make a multi-million-dollar mistake, but it is also very memorable." The "mistake" in this case was in starting a team of 150 programmers on implementation before the system design was complete.

Brooks goes on to suggest solutions for many of the pitfalls in software development, from how to organize a small team on a particular task to how the overall scheduling and management can work. Communications among those developing the software is one of the most important aspects of successfully building a large software system, and Brooks spends a couple of chapters on this topic alone. He also deals with system optimization and testing issues.

New Chapters, New Topics

The anniversary edition is an update from the original text. Rather than revising the text, however, Brooks opted to expand it, adding several new chapters. These include a well-known paper from 1986, "No Silver Bullet," which predicted that software engineering would not experience anywhere near the rate of advance that other aspects of computer technology have enjoyed.

"No Silver Bullet" examines past breakthroughs in software engineering (such as interactive programming, high-level languages, and unified programming environments) and compares them to technologies that were emerging when the paper was written: Ada, object-oriented programming, artificial intelligence, expert systems, automatic program verification, and graphical programming. Brooks correctly predicted that none of these technologies would produce advances for software development at the same rate of advance seen in hardware engineering.

While Brooks' assertion in this essay has proven correct, the paper sparked a significant amount of discussion in the field, which he addresses in a subsequent chapter. In another chapter added for this edition, he summarizes the results of *The Mythical Man-Month* research in a concise outline, occasionally adding commentary.

Hindsight

In the final chapter, "The Mythical Man-Month after 20 Years," Brooks comments on how changes in the field have

affected the results from the original book. Some of the observations have held up quite well, such as the need for conceptual integrity and the importance of an architect to oversee the design of a large project. The assertions raised in the original book, such as the "man-month" being a useless measure, and Brooks' Law also still ring true two decades later.

In looking back, Brooks is most surprised by the explosion of the personal computer field, which has made computing accessible to many more people in many more ways than he foresaw in 1975. It has also had a major impact on software development. For example, the emergence of standardized user interfaces (such as the MacOS, Windows, and Motif) and off-the-shelf, shrink-wrapped packages have dramatically changed how software is developed. Solutions that used to require expensive custom development can now be purchased at a local store. Because of advances in hardware, computer power is rarely even an issue in most software development today. New techniques, such as canned frameworks, allow software to be incrementally "grown" rather than written sequentially from beginning to end.

For anybody involved in large software projects, *The Mythical Man-Month* remains one of the most useful texts in the field. Although it is 20 years old now, many of the observations and results Brooks reported then hold true today. The new edition is a good reason to revisit this classic text.

● **John Peterson** *is a senior developer of Web authoring applications in the Internet products group of Adobe Systems in Mountain View, CA. He can be reached at jp@acm.org.*

To purchase books in this column, contact your local bookseller.

World-Wide Web addresses of *UniForum Monthly* advertisers are listed complimentarily each month.

The ad index is published as a service.
The publisher assumes no liability for errors or omissions.