

# Update

## A Fresh Look at Software Project Estimation: Part I — Promises Impossible to Keep

by E.M. Bennatan, Senior Consultant,  
Cutter Consortium

Software estimation is an area where more promises have been broken than in any other area of software development. But like addicts, we keep demanding promises because we need them to survive. As managers, there is a limit to what we can reasonably expect from software estimation, and when we cross that limit, we are, in effect, asking for short-term satisfaction in the form of promises that will not be kept.

University of Southern California researcher John Lewis questions whether software can be objectively estimated at all and asserts that there are limits that contradict some of the enthusiastic claims made by commercial software estimation advocates.<sup>1</sup> Lewis believes that software development should not be viewed as a manufacturing process, but as a process more akin to research.

At Cutter Consortium, we have been observing the way software organizations estimate software successes and failures, as well as their effects on the organization. The first survey, which was conducted in 2002,<sup>2</sup> exposed software estimation as “a tough beast to control.” According to the findings, 40% of companies surveyed reported poor software schedule and budget estimation, while only 14% reported good performance. Analysis of this unimpressive report card indicated that successful software estimation requires as much of senior management as it does of the developers themselves.

The analysis of the survey concluded with a statement that the beast was alive and well: “Some have learned

to restrain it; some have learned to live with it; and apparently some continue to suffer its wrath.”

Now, six years later, we wondered what has changed. So Cutter again examined the beast to see whether we are now doing any better at controlling it. The latest survey studied software project estimation at more than 100 software development organizations,<sup>3</sup> and the analysis of the results used the same criteria that were used in the earlier survey (see sidebar).

The survey covered a broad range of organizations: 34% have fewer than 25 developers, 23% have between 25 and 100, and 43% have more than 100 developers (of them, 26% have more than 400). At 61% of surveyed companies, projects are typically small — fewer than 5 person-years; at 32%, they are between 5 and 20 person-years (of them, at 10%, they are between 10 and 20); while the remaining 7% typically have larger projects that are more than 20 person-years (of them, at 3%, they are more than 40).

### THE DEFINITION OF SUCCESS

We chose a common measure of success in project estimation; namely, hitting the mark within 10%. This means that if a 20-month project was completed within 18 to 22 months, then it was essentially considered to have been completed on time.<sup>1</sup> We rated development organizations' performance as excellent when more than 90% of projects were completed on time (within 10% of the estimate) and poor when fewer than 50% were completed on time (meaning that most projects were late!).

<sup>1</sup>This, of course, raises the question of how successful project managers are if they deliver a 20-month project early — in, say, 14 months (presumably, a desirable result). Overestimation is still poor estimation, as it can impede efficient resource planning, and can lead to a final product with less functionality than could have been achieved with the full original schedule. Unfortunately, this reasoning is somewhat academic, as overestimation is so rare.

In this *Executive Update*, the first in a series of three comparing the results of the two software estimation surveys, we begin by exploring whether our estimating skills have improved. We then look at the proportion of project failures that are caused by overbudget and over-scheduled projects, and we examine how organizations perceive their own performance in project estimation. Finally, we attempt to determine whether software organizations recognize the limitations of what various estimation methods can realistically achieve.

**HAVE OUR ESTIMATES IMPROVED?**

The first area we examined was comparative performance; how are projects estimated today compared to six years ago? Figure 1 shows the survey findings for schedule estimates. As a reminder, success was defined by the  $\pm 10\%$  rule.

Just 33% of surveyed software organizations report a success rate of more than 70%, which means that more than 70% of their projects were developed on time (of which 8% report a success rate of more than 90%). It is interesting to note that the numbers are not radically

different from the 2002 survey, when 35% reported a 70% success rate (of which 14% had a success rate of more than 90%). However, we do see a certain decline in the schedule estimation success rate, especially in the 90+% success group.

In the mid-to-lower end of the success scale, the differences were more profound. The current survey found that 36% of organizations had a schedule success rate of between 50% and 70% (it was just 26% in 2002), and 24% had a success rate of less than 50% (36% in 2002). Total failure (no projects were on time) was reported by 8% of organizations (3% in 2002).

The latest survey shows a small shift toward the center. This means that while schedule estimates today are not as poor on the low end of the scale as they were in 2002, they are also not as good on the high end.

We next examined the survey findings for budget estimates and compared them to the 2002 findings. According to the data presented in Figure 2, 37% of software organizations had a budget success rate of 70% or more (of which 10% had a greater than 90% success rate). This is surprisingly similar to the 2002

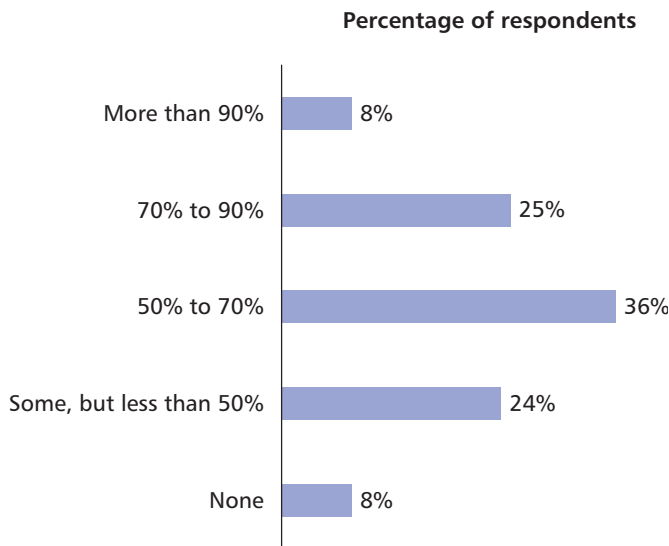


Figure 1 — In the past three years, what would you say is the percentage of your organization’s software projects that are developed within plus or minus 10% of the original estimated time?

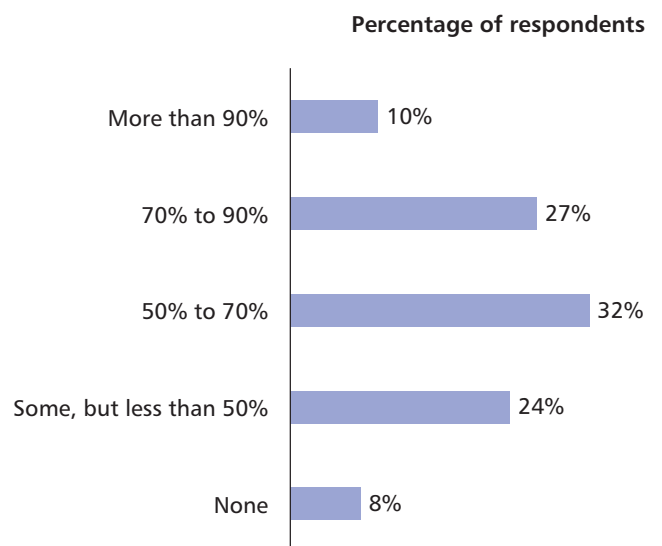


Figure 2 — In the past three years, what would you say is the percentage of your organization’s software projects that are developed within plus or minus 10% of the original estimated budget?

The *Executive Update* is a publication of the Agile Product & Project Management Advisory Service. ©2008 by Cutter Consortium. All rights reserved. Unauthorized reproduction in any form, including photocopying, faxing, image scanning, and downloading electronic copies, is against the law. Reprints make an excellent training tool. For information about reprints and/or back issues of Cutter Consortium publications, call +1 781 648 8700 or e-mail [service@cutter.com](mailto:service@cutter.com).

survey findings, where 38% had a 70+% success rate (of them 10% had a 90+% success rate).

Here, too, the mid-to-lower end of the budget success scale showed more significant differences. The new survey found that 32% had a success rate between 50% and 70% (24% in 2002), and 24% had a less than 50% success rate (34% in 2002). Total budget failure (no project completed within budget) was identical to total schedule failure, and was reported by 8% (4% in 2002). Here again, we found a shift toward the center, similar to the schedule estimate findings.

### ARE ESTIMATE DISASTERS NOW LESS COMMON?

One of the most costly results of poor estimation skills is often the complete cancellation of a project. The survey examined the extent to which software organizations have abandoned or cancelled projects over the past three years, due to significant budget or schedule overruns.

Figure 3 shows that just over half (52%) of surveyed software organizations have had no projects cancelled or abandoned; in the 2002 survey, the figure was 56%. A further 29% report that up to 10% of their projects were cancelled (also 29% in 2002), and 17% state that between 10% and 25% had been abandoned (11% in 2002).

Beyond that, the numbers were minuscule. Just 2% report that the proportion of cancelled projects was

between 25% and 50% (3% in 2002), and no surveyed companies had more than 50% of their projects abandoned (it was 1% in 2002).

It was quite surprising to see that project disasters have actually increased (for companies with more than 10% of project failures). In other words, the performance of poorly performing organizations has declined even further.<sup>4</sup> Other than that, the findings are very similar for both surveys.

### SOFTWARE ORGANIZATIONS' SELF-PERCEPTION

To what extent are the results of the two surveys similarly reflected by the impressions of the software development organizations themselves? We examined this question, and the findings are shown in Figure 4.

Only 37% of software organizations believe that their ability to estimate software projects has improved over the last three years (down from 51% in 2002). A further 58% see no change (up from 43% in 2002), and 5% see a decline (down from 6% in 2002).

It is interesting that software organizations see less improvement of their estimation skills, compared to what they perceived six years ago, and the difference is quite significant. This finding can be interpreted in two ways; either estimation skills are leveling out, with little improvement expected in the future, or software projects are becoming more difficult to estimate.

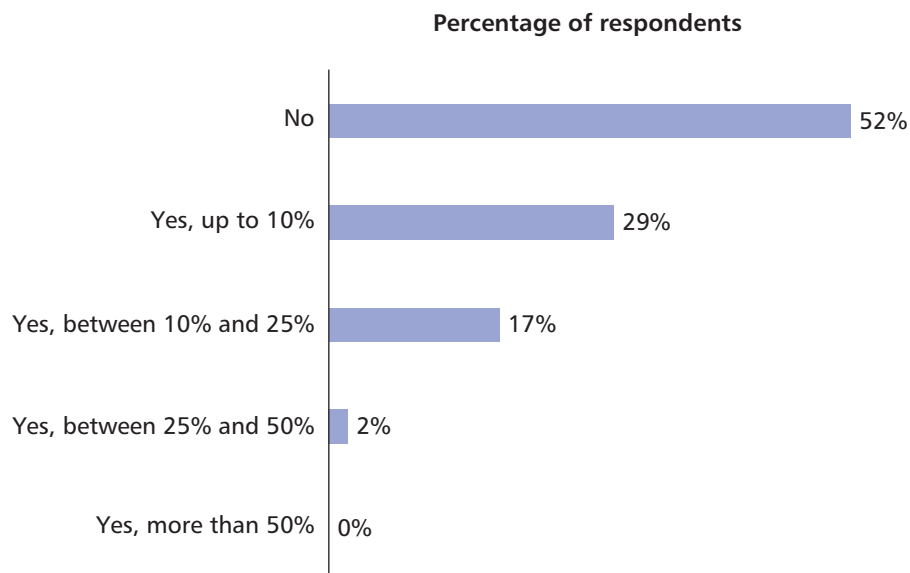


Figure 3 — In the past three years, have any of your organization's software projects been abandoned or cancelled due to significant cost or time overruns?

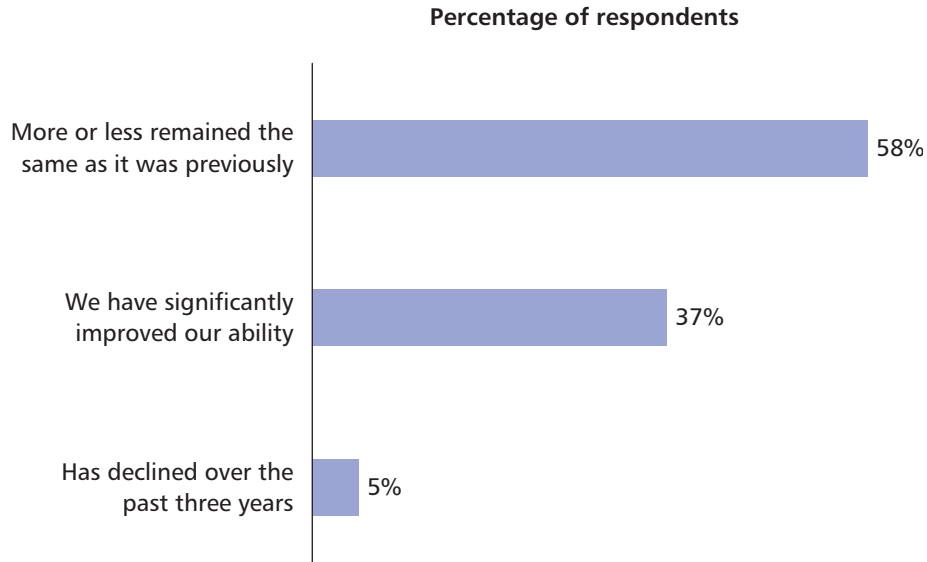


Figure 4 — How would you say your organization’s ability to estimate software project schedules, budgets, and staffing has fared over the past three years?

<b>Table 1</b>				
<b>The performance of 8,000 projects in 350 organizations<sup>1</sup></b>				
	<b>2000</b>	<b>2002</b>	<b>2004</b>	<b>2006</b>
Percentage of projects delivered within budget and schedule	28	34	29	35
Percentage of projects cancelled before completion	23	15	18	19
Percentage of projects over-run on budget and schedule	49	51	53	46

Figure 5 — Boehm and Valerdi’s project estimation performance data.

**SO WHAT ABOUT OBJECTIVE LIMITS?**

Most areas of computer technology have improved immensely in the past six years. In software development, we have learned the benefits of shorter, more agile projects, and even for larger projects, we have adopted a more evolutionary approach. But while new tools, techniques, and methodologies abound, the survey findings do not indicate that they are having any appreciable effect on our ability to estimate.

Interestingly, the Cutter findings correlate rather well with data published recently by University of Southern California’s Barry Boehm et al., which shows no improvement in software project estimation

between 2002 and 2006 (see Figure 5).<sup>5</sup> (As we have seen, the Cutter surveys extend this observation to 2008.)

So why is our ability to estimate software not improving? Have we reached the limit of our abilities? This is a question that is raised by both the Cutter and the Boehm findings, as well as by the perception of software organizations (see Figure 4). All indicate that improvement in software estimation has come to a virtual standstill.

It is difficult, based on the survey data discussed so far, to determine the extent to which software organizations recognize the limitations on what can realistically be

achieved with estimation techniques. But there is much more survey data available that can help us understand this issue.

The next two *Updates* in this series will discuss additional survey data and will continue to examine the question of objective limits on software estimation. We will look at the most common causes of project overruns and the methods used to resolve the most common problems. We will also report on software estimation training and the use of estimation techniques, and we will look at the effects of conflict within the organization derived from the different expectations of various groups.

Readers who would like to comment on this discussion or on their experience with software project estimation are invited to e-mail me at ebennatan@cutter.com.

## ENDNOTES

<sup>1</sup>Lewis, J.P. "Large Limits to Software Estimation." *ACM Software Engineering Notes*, Vol. 26, No. 4, July 2001, pp. 54-59.

<sup>2</sup>Bennatan, E.M. "The State of Software Estimation: Has the Dragon Been Slain? Parts I-III." Cutter Consortium Agile Product & Project Management *Executive Update*, Vol. 3, Nos. 10-12, 2002.

<sup>3</sup>To be statistically accurate, in some cases more than one response may have come from the same organization.

<sup>4</sup>Admittedly, they are likely not the same companies as in 2002.

<sup>5</sup>Boehm, Barry W., and Ricardo Valerdi. "Achievements and Challenges in Cocomo-Based Software Resource Estimation." *IEEE Software*, Vol. 25, No. 5, September 2008, pp. 74-83.

## ABOUT THE AUTHOR

E.M. Bennatan is a Senior Consultant with Cutter Consortium's Agile Product & Project Management practice and a contributor to that advisory service. His extensive hands-on management experience stems from many years as Senior Director at Motorola, Inc., developing large software systems and leading multinational design centers. He has also been VP of Engineering at Midway Corporation, where he managed several hundred software and hardware engineers. A frequent lecturer and speaker on software project management,

Mr. Bennatan is author of the 2006 Jolt Productivity Award-winning *Catastrophe Disentanglement: Getting Software Projects Back on Track* and *On Time Within Budget: Software Project Management Practices and Techniques*, now in its third edition. His books have been translated into several languages, including French, Chinese, Japanese, and Korean. For more by Mr. Bennatan, see Cutter's *Executive Report* "IT Offshore Outsourcing: The Elephant in the Room" as well as his recent *Executive Update* series: "Catastrophe Disentanglement: What Can You Do When Projects Fail?"; "The Latest on Software Project Management"; and "The Project Concept Phase."

His main areas of expertise include software project management, software project catastrophe disentanglement and resolution, introduction of orderly process into ad hoc software organizations, establishing software development organization structure, reengineering and simplification of existing software processes, and management of multinational software development.

Mr. Bennatan is a senior member of the IEEE Computer Society and a member of the ACM. He is currently President of Advanced Project Solutions, Inc. He can be reached at ebennatan@cutter.com.

# SUMMIT 2009



Interact with the Experts in IT

## Now, More Than Ever ...

### Summit 2009

4–6 May 2009 | Cambridge, MA, USA

#### Why Attend Summit 2009?

IT is being challenged to scrutinize every expenditure and make sure that every dollar is invested — not just spent — on high ROI efforts. *Summit 2009* is one of those high-value items. Join us at *Summit 2009* and explore the issues and strategies you should be considering to ensure you're maximizing today's return and keeping ahead of the competition.

#### Summit 2009: Unlike Any Other IT Conference

You'll get advice that is insightful and forthcoming. You'll have a chance to voice your opinion and hear the opinions of international experts and of other IT professionals. You'll debate with, and listen to the debate among, some of the most experienced and clear-thinking business technologists today. And you'll return home with ideas to implement right away. Our only agenda is to support you, to give you a forum for thinking about your business world, outside of the box. Don't miss this opportunity.



Visit [www.cutter.com/summit.html](http://www.cutter.com/summit.html) for an updated agenda



#### The 5 Essential Habits of Appropriately Paranoid Business Technology Strategists

Keynote by Steve Andriole



#### Information Security: Zeroing in on a Moving Target

Keynote by Mark Seiden



#### Case Study: How to Avoid Getting Flattened by a Runaway Project

Taught by Rob Austin



#### Don't Downsize. Rightplace Instead.

Keynote by Vince Kellen



#### Where Do We Go from Here?

Keynote by Tom DeMarco



#### Enterprise Agility: Scaling Up and Out

Seminar with Jim Highsmith



#### Business Architecture: A Case Study

Case Study with Ken Orr



#### Using Operational Business Patterns to Focus Business Architecture

Seminar with Jeroen van Tyn



#### IT Managers Forum

Forum with Warren McFarlan



#### Collaboration for the Enterprise

Seminar with Mike Rosen and John Tibbetts



#### SaaS — How the Cloud Computing Environment Can Work for You

Lunch Discussion with Jeff Kaplan



Interact with the Experts in IT

**Jump ahead of the competition. Reserve your seat for Summit 2009 today!**

**“The Summit stretched my thinking in some new directions.”**

— Martin Malley  
AVP, Information Systems  
Union Pacific Railroad

**“Engaging in a way that other conferences don’t accomplish.”**

— David Smalley  
IT Group Manager  
Progressive

**“Learned some new ways of looking at or managing things.”**

— David Sum  
Director Product Platforms  
Custom House

## Register Today

To ensure a lively and interactive event, attendance for *Summit 2009* will be limited. Register early and guarantee a space for yourself at this exclusive conference.

Return this form by fax to +1 781 648 8707, or mail it to Summit 2009, Cutter Consortium, 37 Broadway, Suite 1, Arlington, MA 02474-5552, USA. Checks should be made payable to Cutter Consortium. To register by phone, call +1 781 648 8700. Or visit [www.cutter.com/summit.html](http://www.cutter.com/summit.html) to register online.

**# Registrants    Total**

**Summit 2009 Conference** (4–6 May 2009)

US \$1,995 each

\_\_\_\_\_ \$ \_\_\_\_\_

If you would like to register your team of 4 or more for Summit 2009, please contact your account representative for multiple-attendee discounts.

\_\_\_\_\_  
Name Title

\_\_\_\_\_  
Organization Department

\_\_\_\_\_  
Address Mail Stop/Suite #

\_\_\_\_\_  
City State/Province

\_\_\_\_\_  
ZIP/Postal Code Country

\_\_\_\_\_  
Telephone E-Mail

Check enclosed (payable to Cutter Consortium).

Please invoice my company.

Charge MasterCard, Visa, AmEx, or Diners Club. (Charge will appear as Cutter Consortium.)

\_\_\_\_\_  
Account Number Expiration Date

\_\_\_\_\_  
Name on Credit Card Signature

\_\_\_\_\_  
Registrant #1 Name Title

\_\_\_\_\_  
Registrant #2 Name Title

\_\_\_\_\_  
Registrant #3 Name Title

\_\_\_\_\_  
Registrant #4 Name Title

Registrations for *Summit 2009* are transferable. If you find that you cannot attend, you may transfer the registration to another member of your organization at any time prior to the first day of the event. Cancellations after 31 March 2009 are nonrefundable. In the unlikely event that Cutter Consortium cancels the conference, Cutter Consortium is not responsible for any airfare, hotel or other costs incurred by registrants. Speakers subject to change without notice.

60R\*C11FLY