A Replicated Survey of IT Software Project Failures

Khaled El Emam, *University of Ottawa*

A. Güneş Koru, *University of Maryland, Baltimore County*

Results from our global Web survey of IT departments in 2005 and 2007 suggest that, although the overall project failure rate is high, word of a software crisis is exaggerated.

The prevailing view that there's a software crisis arose when the Standish Group published its 1994 Chaos report,[1] which indicated a high cancellation rate for software projects. But a number of practitioners and researchers have questioned its findings[2,3] and criticized it for not disclosing its methodology, lack of peer review, inconsistent reporting, and misconceptions about the definition of failure.[4] Since 1994, other researchers have published evidence on project cancellation (see the sidebar). However, this evidence is somewhat equivocal because many of these studies weren't peer reviewed and didn't publish their methodologies, which makes judging the evidence's quality difficult.

So, the software community still needs a reliable global estimate of software project cancellation rates that will help us determine whether there's a software crisis. Knowing the project cancellation rate will let software organizations benchmark their performance to see how they compare to others in the industry.

We conducted a replicated international Web survey of IT departments (developing management information systems) in 2005 and 2007. We aimed to

- estimate IT projects' actual cancellation rates,
- determine what factors have the biggest impact on cancellation,
- estimate the success rate of projects that deliver, and
- determine whether project size affects cancellation rates and project success.

Methods

Project cancellations aren't always a bad thing. Cancelled projects could lead to substantial learning or produce artifacts applicable to future projects.[5] Nonetheless, project cancellations waste corporate resources, and they are often difficult to deal with because they require special management skills and critical business decisions.[6]

We also need to consider that project cancellation and performance might depend on size. Smaller projects tend to have lower cancellation rates,[7–9] and of projects that deliver, smaller projects tend to perform better in terms of quality, being on budget, and being on schedule.[7,10]

Measurement

The unit of measurement for this study was the software project (not process improvement, organizational change, or business-process-reengineering projects). We measured cancellation, project size, and project performance.

0740-7459/08/$25.00 © 2008 IEEE

What Do We Know about Software Project Cancellation Rates?

Table A summarizes the existing evidence. There's a general decreasing trend over time, with the most recent estimates mostly below the 20-percent level. However, there's variation within a narrow range.

We also wanted to determine the software community's perceptions of the software project cancellation rate to see if it matched the evidence. We sent a brief email survey to *IEEE Software*'s 2006 reviewers to determine their perceptions of the average cancellation rate for software projects. The response rate was 37.33 percent (84 out of 225 targeted), which is consistent with other electronic surveys.[11] Figure A shows the responses' distribution.

The largest cancellation rate category was 11 to 20 percent, estimated by 27.38 percent of the respondents. This is consistent with evidence from recent studies in Table A. However, 60.71 percent of respondents estimated the average cancellation rate to be above 20 percent, which is higher than the evidence in Table A would suggest. There was a wide spread for those responses, with some estimating the average cancellation rate to be above 50 percent.

So, in summary, existing evidence is consistent and shows a decreasing trend, and the community perception of cancellation rates tends to be higher than the evidence. This suggests a gap in our understanding that requires some bridging.

Table A

A summary of evidence on software project cancellation rates*

Study, year, and location	Cancellation/abandonment rate (%)
Standish Group, 1994, US	31
Standish Group, 1996, US	40
Standish Group, 1998, US	28
Jones,[8] 1998, US (systems projects)	14
Jones,[8] 1998, US (military projects)	19
Jones,[8] 1998, US (other projects)	> 24
Standish Group, 2000, US	23
Standish Group, 2002, US	15
Computer Weekly,[9] 2003, UK	9
UJ,[10] 2003, South Africa	22
Standish Group, 2004, US	18
Standish Group, 2006, US	19

**The Standish Group data comes from various reports.[1–7]*

References

1. "Latest Standish Group Chaos Report Shows Project Success Rates Have Improved by 50%," Standish Group, 2003.
2. *2004 Third Quarter Research Report*, Standish Group, 2004.
3. *Extreme Chaos*, Standish Group, 2001.
4. *Chaos Report*, Standish Group, 1994.
5. *Chaos: A Recipe for Success*, Standish Group, 1999.
6. J. Johnson, "Turning Chaos into Success," *Softwaremag.com*, Dec. 1999, www.softwaremag.com/L.cfm?doc=archive/1999dec/Success.html.
7. D. Rubinstein, "Standish Group Report: There's Less Development Chaos Today," *Software Development Times*, 2007.
8. C. Jones, "Project Management Tools and Software Failures and Successes," *Crosstalk*, July 1998, pp. 13–17.
9. C. Sauer and C. Cuthbertson, "The State of IT Project Management in the UK 2002–2003," *Computer Weekly*, 15 Apr. 2003.
10. R. Sonnekus and L. Labuschagne, "Establishing the Relationship between IT Project Management Maturity and IT Project Success in a South African Context," *Proc. 2004 PMSA Global Knowledge Conf.*, Project Management South Africa, 2004, pp. 183–192.
11. M. Schonlau, R.D. Fricker, and M.N. Elliott, *Conducting Research Surveys via Email and the Web*, Rand, 2002.

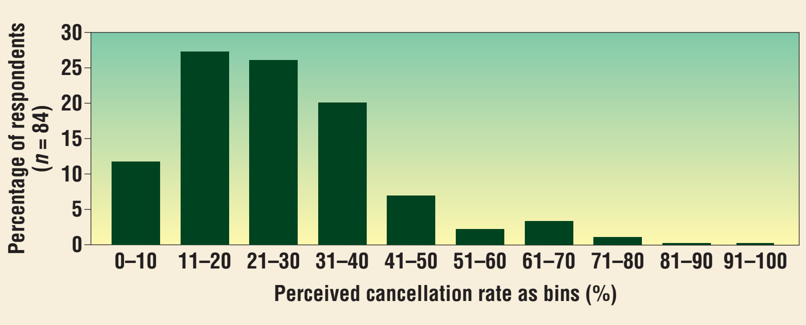

Figure A. Survey respondents' perceived cancellation rates (*n* = 84). This gives the software community's perceptions about where the industry stands today and serves as a useful comparison to the results of our survey of actual projects.

We measured cancellation using a binary question. We considered a project to be cancelled if it didn't deliver any usable functionality by the first release. (This is consistent with previous definitions, which considered a project cancelled if it was cancelled before completion or delivered functionality that wasn't used.[1])

We measured size in terms of the duration to

Table 1
The distribution of companies where the projects were performed*

Year	Domain	Percentage
2005	Financial services	22
	Computer consulting and systems integration	21
	Computer software publisher	12.5
	Government (nonmilitary)	9
	Telecommunications	5
	Outsourcing/Web services	5
2007	Computer consulting and systems integration	19
	Financial services	18
	Computer software publisher	12
	Government (nonmilitary)	6
	Medical and health services	4.5
	Colleges and universities	4.5

We only show the largest business domain categories.

Table 2
The distribution of respondents by their title in 2005 (n = 232) and their role in 2007 (n = 156)*

Year	Role or title	Percentage
2005	Project manager	26
	External consultant	16
	Director	14
	Software engineer	9
	Quality assurance management	7
	Vice president/chief executive officer	6
	Other	22
2007	Project manager	26
	External consultant	24
	Architect or lead technical role	13
	Sponsor	9
	Developer	5
	End user or end-user representative	5
	Other	18

In the intervening years, the Cutter Consortium changed how it characterized its clients, causing the respondents' characterization to change.

first release in months (henceforth "duration") and peak staffing. We counted duration from the project's initial concrete conception (for example, when a formal business requirement is made and a project approved). For both variables, the respondents chose among different size intervals. For cancelled projects, we measured duration until the project cancellation date.

For projects that weren't cancelled, we measured project performance using five success criteria: user satisfaction, ability to meet budget targets, ability to meet schedule targets, product quality, and staff productivity. We used a four-point Likert scale (excellent, good, fair, and poor) for the response categories. This type of subjective performance measure has been used extensively in the past.[11,12]

In the 2007 survey, we also collected data on the reasons for cancellation.

Data collection

We collected the data for this study through a Web survey of Cutter Consortium (www.cutter.com) clients. We targeted midlevel and senior-level project managers in IT departments who would have firsthand knowledge and involvement in software projects. We sent email invitations with reminders for nonrespondents, both in spring 2005 and 2007. We asked respondents to complete the questions for the most recent project that they worked on.

We obtained 232 responses for the 2005 survey and 156 responses for the 2007 survey. Because the response rate would reveal the size of the Cutter Consortium's customer base, we can't report it. However, to gauge whether there was a nonresponse bias, we statistically compared the early responses with the late ones for all variables[13] using a Mann-Whitney U test.[14] Any difference would point to the possibility that if the nonrespondents actually had responded, the results could be different. There was no statistically significant difference between the early and late responses for any of our variables in either 2005 or 2007 using a two-tailed test at an alpha level of 0.05. So, there's no evidence of nonresponse bias.

Data analysis

We present cancellation rates descriptively (as a percentage) with 95-percent confidence intervals.

We examined the relationship between cancellation and size using ridit analysis.[15] This is suitable when evaluating the relationship between a categorical and binary variable. The ridit value is the probability that a randomly selected cancelled project is larger than a randomly selected completed project as measured by duration or peak staffing. If the ridit value is equal to 0.5, there is no relationship.

To investigate the relationship between size and project performance (for delivered projects), we used the Gamma statistic.[14] This is a form of correlation coefficient for ordered (for example, questionnaire) variables. We performed statistical tests at a Bonferroni-adjusted alpha level of 0.05. This approach adjusts for finding spurious correlations when we do many statistical tests.

Results

Here, we present our descriptive results. We then examine the project cancellation rate and the performance of delivered projects that weren't cancelled.

Descriptive results

Table 1 shows the distribution of companies in the 2005 and 2007 surveys, and Table 2 shows the individual respondents according to their titles and roles. Financial services, consulting and systems integration, and software publishing companies represented the largest business domains covered. The largest groups of respondents were project managers and external consultants (who had key responsibilities on the projects).

In the 2005 survey, the largest number of respondents came from the US (37 percent), followed by Australia (11 percent), the UK (8 percent), and Canada (4 percent). In the 2007 survey, the largest number of responses came from the US (38 percent), followed by India (14 percent), Canada (10 percent), and the UK (6 percent). Compared to the 2005 numbers, the percentages for Canada and India were higher, further flattening the global distribution.

Figure 1a and Figure 1b show the distribution of the projects as measured by duration and peak staffing. There's considerable consistency in the distributions across the two years.

Approximately half the projects lasted nine months or less. So, there was no clear preponderance of short projects versus long projects. The most common number of developers was between three and 10. Most projects had fewer than 10 developers.

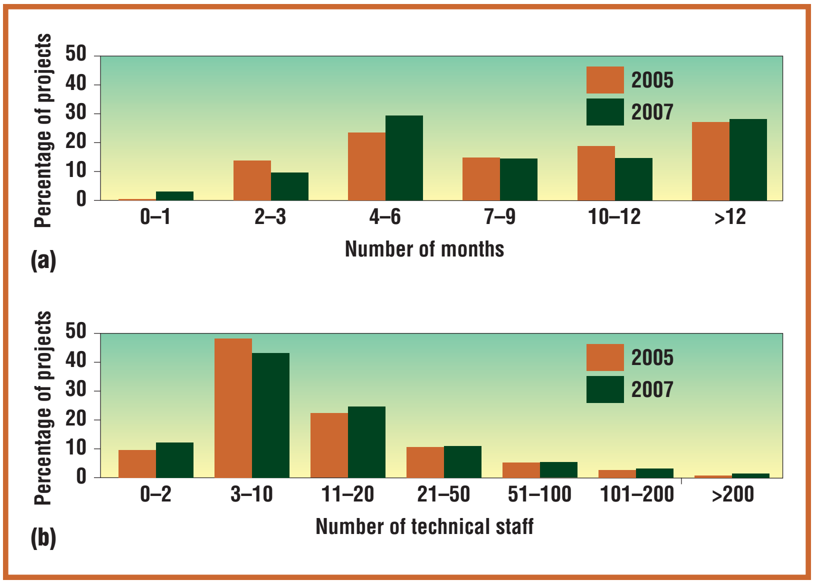

Figure 1. The distribution of projects according to (a) duration (in months) and (b) peak staffing (number of technical staff). This will help us interpret the later results showing project outcomes.

Project cancellations

Of all the projects, 15.52 percent were cancelled in 2005 and 11.54 percent were cancelled in 2007, before they delivered anything. This decrease over time was not statistically significant (p = 0.19 using a binomial test).

Our ridit analysis found no statistically significant difference in the cancellation probability for project duration or peak staffing. Figure 2 shows the ridit plots for 2005 and 2007.

As we mentioned before, our 2007 survey also asked about the reasons for project cancellation. Table 3 (see p. 88) summarizes the results. The two most common reasons were requirements and scope changes and lack of senior management involvement, both chosen by 33 percent of the respondents. These were followed by budget shortages and lack of project management skills, both chosen by 28 percent of the respondents.

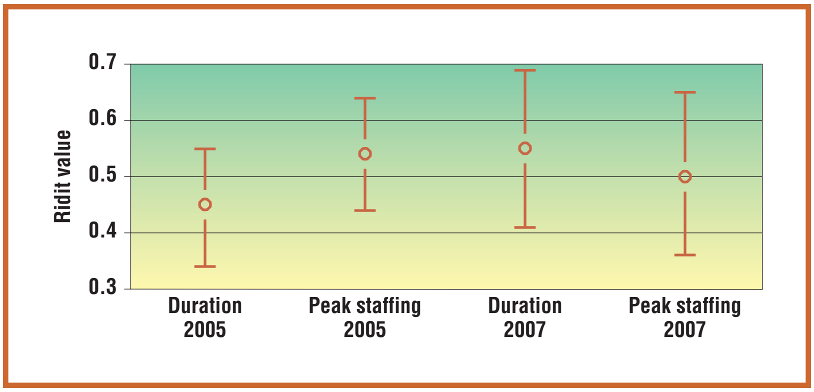

Figure 2. How project duration and peak staffing affected cancellation rates (2005 and 2007). The plots show the ridit score (represented by a dot in the middle of the bar) with the 95-percent confidence interval (the length of the bar). All confidence intervals intersect with the 0.5 line, which means that there's no duration or peak-staffing difference between the cancelled and delivered projects.

Project success

Figure 3 (see p. 88) classifies the project performance responses as a failure if the respondents rated them "poor" or "fair." Most respondents felt that user satisfaction, product quality, and staff productivity were either good or excellent. Approximately one-third or fewer of the respondents perceived that the project success was either poor or fair for those three criteria.

In 2005, approximately half of the respondents characterized their projects' ability to meet budget and schedule targets as fair or poor. In 2007, around 37 percent rated the ability to meet budget targets as fair or poor, and 47.10 percent rated "meeting

Table 3

Reasons for project cancellation with percentages and 95% confidence intervals for the 2007 respondents (n = 18)*

Reason for cancellation	Percentage of respondents (95% confidence interval)
Senior management not sufficiently involved	33 (13, 59)
Too many requirements and scope changes	33 (13, 59)
Lack of necessary management skills	28 (10, 54)
Over budget	28 (10, 54)
Lack of necessary technical skills	22 (6, 48)
No more need for the system to be developed	22 (6, 48)
Over schedule	17 (4, 41)
Technology too new; didn't work as expected	17 (4, 41)
Insufficient staff	11 (1, 35)
Critical quality problems with software	11 (1, 35)
End users not sufficiently involved	6 (0, 27)

*The 95% confidence interval is usually wide because we're looking at only 18 cancelled projects. The respondents had the option of adding qualitative information as well as the predefined categories.

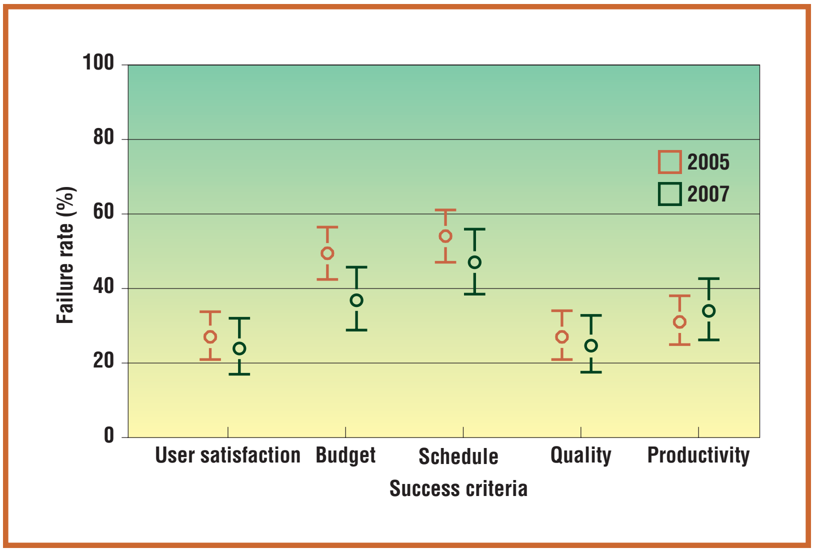

Figure 3. The percentage of respondents rating their delivered project "poor" or "fair" on success criteria, with 95-percent confidence intervals. This indicates which project outcomes were perceived to be the most challenging and how that has changed (or not) over time.

schedule targets" as fair or poor. As we can see, 2007 saw a marked improvement in meeting budget targets. The most critical performance problem in delivered software projects is therefore estimating the schedule and managing to that estimate.

To get a more holistic view of each delivered project's performance, we counted the performance criteria for which the responses were poor or fair. Then, we categorized the projects as *successful* (rated poor or fair on zero or one performance criterion), *challenged* (poor or fair on two or three performance criteria), or *unsuccessful* (poor or fair on four or five performance criteria). Figure 4 shows the percentages and 95-percent confidence intervals for these categories. The results for both years were very similar, with little change over time.

Between 48 percent and 55 percent of delivered projects were considered successful, whereas between 17 percent and 22 percent were considered unsuccessful. A binomial test of the difference between 2005 and 2007 in the proportion of unsuccessful projects wasn't statistically significant, so there's no evidence that the actual failure rate of completed projects has decreased. The combined cancellation plus unsuccessful project rate was approximately 34 percent in 2005 and 26 percent in 2007.

We evaluated whether project duration and peak staffing load were correlated with any of the five success criteria. Correlation analysis results indicate a statistically significant and moderately sized Gamma correlation[14] between duration and budget for 2005 ($\gamma = 0.189$) and 2007 ($\gamma = 0.243$). We found a statistically significant correlation in 2005 between duration and schedule ($\gamma = 0.188$), as well as duration and productivity ($\gamma = 0.197$). We didn't find a significant correlation between peak staffing and any of the success criteria.

Discussion

Project failure/success surveys help the community understand the status of software development outcomes and can be useful for benchmarking purposes. In addition to presenting up-to-date data, this study attempted to address some of the criticisms of previous work in this area.

Project failure rates

The IT project cancellation rate ranged from 11.5 to 15.5 percent. Despite the differences in the type of projects and methodology, these numbers don't differ much from the results of recent surveys shown in the sidebar. Our results are also consistent with the mode response from the perceptions of community experts.

The cancellation rates for the 2005 and 2007 surveys were similar, with no statistically significant difference, indicating that no improvement occurred during these two years. However, since 1994, there has been a clear trend of decreasing cancellation rates across all studies, despite the differences in types of projects and methodologies. It's unlikely that the cancellation rate will converge to zero someday because developers will always need to cancel projects, even if only owing to changes in business needs. It's an empirical question whether we have reached a plateau, however.

We didn't find a relationship between project size and the cancellation rate. These findings contradict a common software engineering belief that shorter projects will less likely be cancelled, perhaps because their scope is usually small and there's less communication complexity among the team members. You also could argue that the significant sunk investment in longer projects makes cancelling them more difficult—it's easier to cancel a recent project or one with little investment. Previous studies only presented their results descriptively and didn't evaluate whether the observed differences were likely by chance, which is one plausible explanation for other previous studies finding a project size effect.

Changes in requirements and scope were primary reasons for project cancellation. You could argue that if the business requirements change and a system is no longer needed, you should cancel the project. So, this reason doesn't necessarily mean that an inability to manage changes drove cancellation. Going over budget, however, was a key reason why projects were cancelled. Surprisingly, given that many respondents were in management positions, the lack of senior management commitment and inappropriate management skills were also key reasons for project cancellation. The former suggests a misalignment between IT and the business. Poor project management skills can cause and exacerbate all these problems.

Meeting schedule targets was consistently the most challenging outcome for delivered software projects. This highlights the importance of transitioning better estimation techniques into projects. Shorter projects tended to have a greater chance of meeting budget targets, with equivocal evidence on their ability to meet schedule targets and have high productivity.

Between 16 and 22 percent of delivered projects were considered unsuccessful on the basis of their performance. This is a relatively large number for projects that management decided not to cancel. The combined rate of cancelled plus unsuccessful projects was between 26 percent and 34 percent. By most standards, this would be considered a high failure rate for an applied discipline.

Limitations

One important limitation of this study is its representativeness. The Cutter Consortium client base is likely to contain organizations interested in learning about and implementing good software engineering practices. So, the sampled projects will more likely perform better compared to the whole software engineering industry.

Another limitation of doing a survey with this subpopulation is that the projects are mainly run in IT departments. This group deals with few, if any, real-time embedded systems, for example. You could argue that non-IT projects may have a different cancellation and failure profile. Furthermore, our respondents included few large projects, which could be related to the Cutter Consortium focus on agile practices, which tend to be deployed on smaller projects. Finally, most respondents were from Western countries; consequently, to the extent that national culture helps determine project success, the results might be different for projects with different cultural traditions.

Many factors can affect cancellation rates and project success. Organizational maturity, methodology, and project management experience will affect project success. However, our limited objective was to get an overall aggregate value across IT projects in the software industry. Had we segmented projects by the these factors, we would have seen more variation, with some types of projects performing better or worse than the numbers presented here.

One concern with the project performance variables is that specific respondent roles could be biased toward certain project outcomes. Should that be the case, you could justifiably question our findings. For example, the project manager might be inclined to inflate the project's success compared to the end user or sponsor.

To check for possible role-specific bias in the project outcome responses, we tested whether responses differed on the five project outcome variables (satisfaction, budget, schedule, quality, and productivity) among different role types. For the 2005 survey, we compared internal versus external employees (consultants); for the 2007 survey, we compared internal versus external employees and technical staff versus user staff. We used a nonparametric Mann-

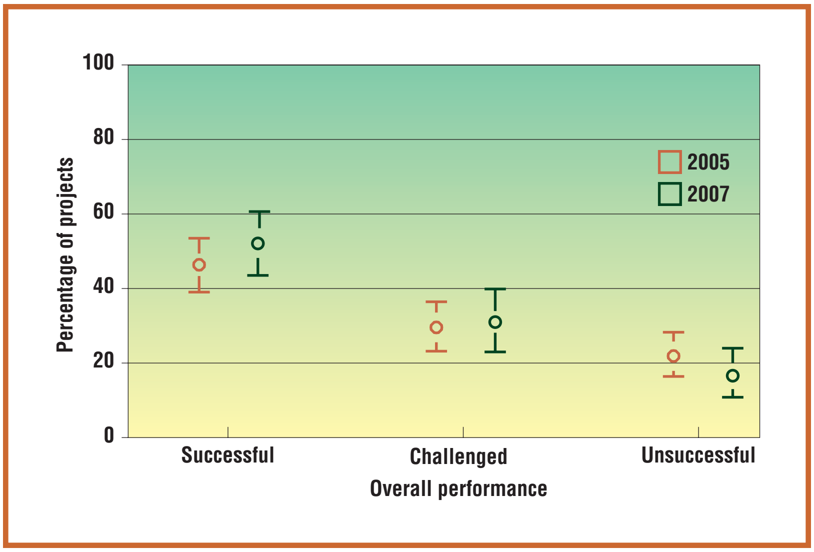

Figure 4. Categorization of projects by their overall performance, showing percentages and 95-percent confidence intervals. This provides an overall success rate for delivered projects, as well as global estimates of unsuccessful delivered projects.

Whitney U test.[15] In all cases, we found no statistically significant difference (alpha = 0.05, 2-tailed). This indicates that there's no evidence of role-specific biases in the outcome measures.

Considering IT software projects only, our results suggest that most projects actually deliver. Talk of a software crisis with the majority of projects cancelled is perhaps exaggerated. If we consider overall failures (cancelled projects plus delivered projects with unsuccessful performance), the most up-to-date numbers indicate that 26 percent to 34 percent of IT projects fail.

There's clearly room for improvement because the overall project failure rate is high for an applied discipline. Despite many years of research, estimation skills remain a key challenge to IT projects either because practitioners aren't using the best estimation tools and techniques available or because the best tools and techniques available require further improvements before practitioners can use them effectively.

Acknowledgments

We thank the Cutter Consortium, all survey respondents, and Robert Glass and Dennis Frailey for their feedback.

References

1. *Chaos Report*, Standish Group, 1994.
2. R.L. Glass, "IT Failure Rates—70 Percent or 10–15 Percent?" *IEEE Software*, vol. 22, no. 3, 2005, pp. 110–112.
3. R.L. Glass, "Failure Is Looking More Like Success These Days," *IEEE Software*, vol. 19, no. 1, 2002, pp. 103–104.
4. M. Jorgensen and K. Molokken-Ostvold, "How Large Are Software Cost Overruns? Critical Comments on the Standish Group's Chaos Reports," *Information and Software Technology*, vol. 48, no. 4, 2006, pp. 297–301.
5. K.R. Linberg, "Software Developer Perceptions about Software Project Failure: A Case Study," *J. Systems and Software*, vol. 49, nos. 2–3, 1999, pp. 177–192.
6. C.L. Iacovou and A.S. Dexter, "Surviving IT Project Cancellations," *Comm. ACM*, vol. 48, no. 4, 2005, pp. 83–86.
7. C. Jones, *Software Assessments, Benchmarks, and Best Practices*, Addison-Wesley, 2000.
8. C. Sauer and C. Cuthbertson, "The State of IT Project Management in the UK 2002–2003," *Computer Weekly*, 15 Apr. 2003.
9. "Extreme Chaos," Standish Group, 2001.
10. C. Jones, "Project Management Tools and Software Failures and Successes," *Crosstalk*, July 1998, pp. 13–17.
11. D.R. Goldenson and J.D. Herbsleb, *After the Appraisal: A Systematic Survey of Process Improvement, Its Benefits, and Factors That Influence Success*, tech. report CMU/SEI-95-TR-009, Software Eng. Inst., Carnegie Mellon Univ., 1995.
12. K. El Emam and A. Birk, "Validating the ISO/IEC 15504 Measure of Software Requirements Analysis Process Capability," *IEEE Trans. Software Eng.*, vol. 26, no. 6, 2000, pp. 541–566.
13. J.R. Lindner, T.H. Murphy, and G.E. Briers, "Handling Nonresponse in Social Science Research," *J. Agricultural Education*, vol. 42, no. 4, 2001, pp. 43–53.
14. S. Siegel and J. Castellan, *Nonparametric Statistics for the Behavioral Sciences*, McGraw-Hill, 1988.
15. S. Selvin, *Statistical Analysis of Epidemiologic Data*, Oxford Univ. Press, 1996.

About the Authors

Khaled El Emam is an associate professor at the University of Ottawa's Faculty of Medicine and the School of Information Technology and Engineering. He's also a Canada Research Chair in Electronic Health Information at the university. In 2003 and 2004, the *Journal of Systems and Software* ranked him the top systems and software engineering scholar worldwide on the basis of his research on measurement and quality evaluation and improvement (he was ranked second in 2002 and 2005). El Emam received his PhD from King's College at the University of London. Contact him at kelemam@uottawa.ca; www.ehealthinformation.ca.

A. Güneş Koru is an assistant professor in the Department of Information Systems at the University of Maryland, Baltimore County. His research interests include software quality, measurement, maintenance, and evolution, with an emphasis on open-source software, Web-based applications, healthcare information systems, and bioinformatics. Koru received his PhD in computer science from Southern Methodist University. He's a member of the IEEE and ACM and an active contributor to the Promise software data repository. Contact him at gkoru@umbc.edu.