Requirements Engineering: The State of the Practice

Colin J. Neill and Phillip A. Laplante, *Penn State University*

Many authors refer to dominant, prevalent, or common practices, approaches, and techniques used in the software development industry. We even found ourselves using such references. When we tried to support those statements with citations, however, we found little applicable data.

Bill Curtis and his colleagues reported the first significant field survey of practices in 1988.[1] More recent investigations include those by Khaled El Emam and Nazim Madhavji in 1995, an investigation of 60 cases;[2] Uolevi Nikula and his colleagues in 2000, a survey of 15 participants from 12 organizations in Finland;[3] and Didar Zowghi and his colleagues in 2001, interviews and questionnaires conducted within a single Australian organization.[4]

During our literature search we did, nevertheless, uncover some useful vehicles for determining current practices: two Web-based surveys on which to model our own—one hosted by Macquarie University in Sydney, Australia, and the other by the University of Calgary.[5] We couldn't find results for the Australian survey, but the Calgary team recently presented the results from 25 completed surveys.[6] Our survey resembles both, with modifications to handle the practice area of each company and project.

Little contemporary data exists to document actual practices of software professionals for software requirements elicitation, requirements specification document development, and specification validation. This exploratory survey and its quantitative results offer opportunities for further interpretation and comparison.

Survey design and conduct

We created a Web-based survey (www.personal.psu.edu/cjn6/survey.html) consisting of 22 questions (summarized in Table 1). We drew our survey participants from a database of prospective, current, and past graduate students of the Penn State Great Valley School of Graduate Professional Studies. We sent them an email invitation (and subsequent reminder) to visit our Web site. This closed invitation ensured that

0740-7459/03/$17.00 © 2003 IEEE

Table 1

Summary of survey questions

No.	Question
1	What type of business or organization are/were you employed by during this project?
2	Approximately how many software professionals are/were employed by your organization?
3	What is/was the approximate size of your organization's annual budget?
4	Which of the following application domains does/did this project apply to?
5	Which of the following development life cycles best describes the one you are using/did use?
6	Within the life cycle, do/did you do any prototyping?
7	If your answer is Yes, how do/did you prototype?
8	How many full-time staff (IT) are/were involved in the project altogether?
9	How many full-time staff are/were involved in each phase of the project development?
10	What is/was the duration of the project (from inception to delivery)?
11	How is/was the project duration distributed among the following phases?
12	What techniques do/did you use for requirements elicitation?
13	Which of the following approaches are you using/did you use in analysis and modeling of the software requirements?
14	In what sort of notation is/was the requirements specification expressed?
15	Do/did you perform requirements inspections?
16	If your answer is Yes, which technique do/did you use?
17	In your opinion, does your company do enough requirements engineering?
18	When you review/reviewed requirements, which of the following approaches do/did you employ?
19	Indicate with a number the size of requirements specification in terms of [the following].
20	The following statements are indicators for Software Quality and Software Productivity. Please rate these statements by clicking one box with the following scales.
21	Which of the following best describes your position while engaged in this project?
22	Over the last five years, how many software projects have you worked on?

we included only industrial practitioners in the survey population because the graduate school caters only to working professionals.

We collected survey data through March and April 2002, and although we received a few additional responses after this date, we didn't include them because the analysis had already commenced. So, of the 1,519 invited persons, we had 194 completed responses.

Participant characteristics

The survey response captured a diverse mix of industries, including Fortune 500 and multinational corporations, consistent with the mix of students surveyed. These companies include Lockheed Martin Management and Data Systems, Merck, SAP, Unisys, Wyeth-Ayerst, M&M Mars, Vanguard, Boeing, AstraZeneca, Dupont, Siemens Medical Systems, Verizon, GlaxoSmithKline, and numerous smaller companies.

The survey participants also reflect a diverse range of positions and experience, as summarized in Figure 1. Although almost three quarters of the respondents identified themselves as system designers, project managers, or technical specialists, almost 20 percent are at the executive, architect, or consultant level, which is again consistent with Penn State Great Valley's student population.

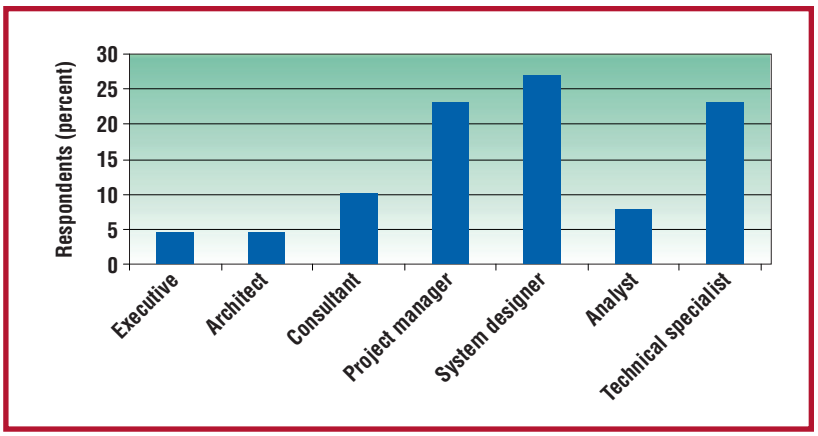


Figure 1. Respondents' positions.

As Figures 2a and 2b show, the survey drew a response from professionals from a wide range of industries and application domains.

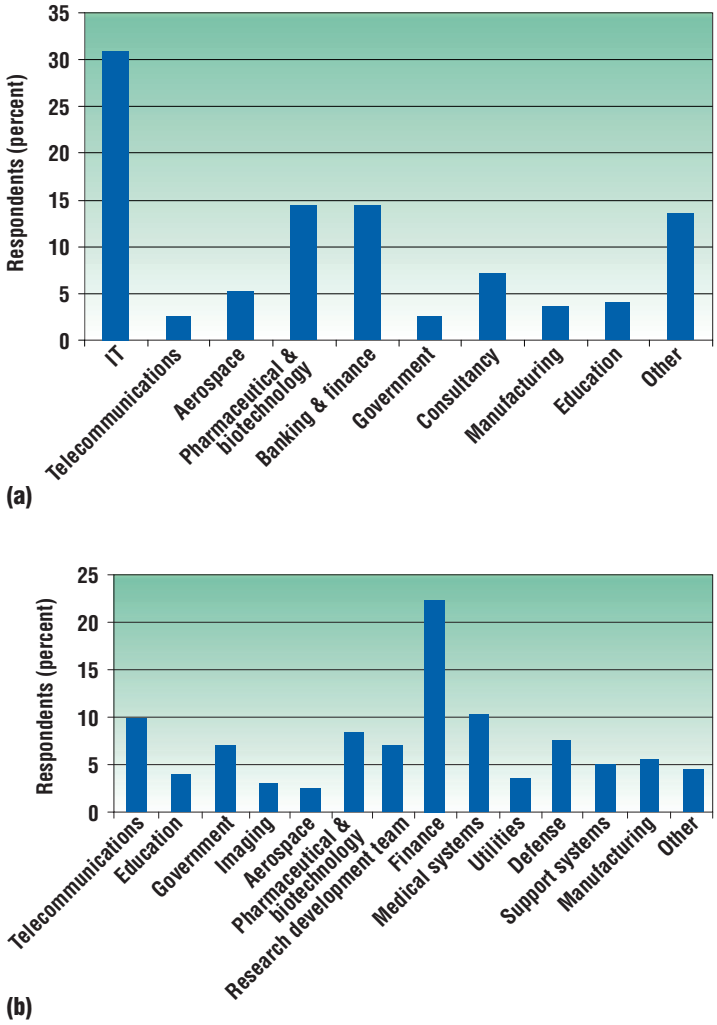


Figure 2. Respondents: (a) organization types and (b) project domains.

Summary of responses

We provide the main survey results in graphical format for brevity. We discuss key trends and noteworthy items throughout, leaving others for the conclusions. We aren't passing judgment on the findings, however, merely presenting them.

Perhaps the most interesting finding we uncovered in this study was the maintained popularity of the waterfall lifecycle model—overall, 35 percent of respondents indicated that they use it. Further analysis linked this question's response to both the respondent's position and the project's duration (see Figures 3a and 3b).

In Figure 3a, we see that the waterfall model is the most prevalent in both distributions. However, the managerial and advanced technical participants felt that they were following fewer waterfall processes, with more evolutionary and incremental process models, than the developers and analysts.

We see a more stark contrast in the distinction between life cycles based on project duration. We found that in projects lasting longer than two years, the incremental life cycle was most common—almost twice as common as in shorter projects.

Given the waterfall model's seeming popularity, the response to the question "Within the life cycle, did you do any prototyping?" was particularly noteworthy. Remembering that the standard waterfall model doesn't involve prototyping, we found that almost 60 percent of respondents did perform some sort of prototyping. Of those who use prototypes, half claimed to apply evolutionary prototyping, where the prototype becomes part of the final production system (see Figure 4).

The survey then focused on the techniques used for requirements elicitation, representation, and modeling. We presented an extensive list of known techniques; participants could select all that applied. The data (see Figure 5a) revealed that over 50 percent surveyed used scenarios or use cases[7] in the requirements phase. This was the most common approach when any approach was used. Contrast this with the fact that a later question revealed that only 30 percent of the survey population reported using object-oriented analysis—a technique often applied along with use cases. We expected a higher proportion given the emphasis on OO in tools, languages, and books popular in the industry. Nevertheless, the survey provided evidence that an OO methodology in requirements specification correlates with systems that are "easier to use" than those in which a structured methodology has been employed (76 percent versus 61 percent). In Figure 5b, 33 percent indicated that they used no methodology at all for requirements analysis and modeling. However, we learned that 70 percent of those felt that the finished product's capabilities fit well with customers' needs, and 63 percent felt that end users found finished products easy to use.

Other popular requirements approaches include group-consensus-type techniques such as User-Centered Design,[8] Joint Application Design,[9] interviews, and focus groups.[10]

We also highlight that we still find considerable use of structured approaches such as the Structured Analysis and Design Technique and Structured Systems Analysis and Design Methodology. For more information

on any of these techniques, we direct readers to Ian Sommerville's general text on software engineering.[14]

The next few questions in the survey focused on the requirements representation's degree of formality and how those requirements were managed. In Figure 6a, we see that formal representations are rare (7 percent reported). However, 51 percent of respondents reported informal representations (for example, natural language). We then wondered whether this lack of formality impacted end-product quality. To answer this, we compared the responses to two questions regarding product suitability and usability ("End users found the finished product was easy to use" and "Capabilities of the finished product fit well with customer or user needs"). Only 69 percent of respondents who used formal representations agreed with the first statement—compared to 79 percent that indicated informal and 76 percent that indicated semiformal. We saw similar results for the second statement: 61.5 percent who reported formal agreed with the statement, 70 percent informal, and 73 percent semiformal.

An interesting corollary to the formality discussion is the subsequent requirements review and inspection effort. We found that 59 percent are performing inspections, apparently employing a range of techniques (see Figure 6b).

Given all the questions regarding requirements engineering practices, we wanted to know how satisfied the participants were with their organization's effort. Fifty-two percent of respondents didn't think that their company did enough requirements engineering—only 29 percent felt that their organization's requirement effort sufficed.

Project success indicators

The statements quoted in the previous section regarding end-product usability and suitability were two of 11 such questions where participants could rate how strongly they agreed or disagreed with the statement. Rather than presenting and discussing the results to each of these questions, we further analyzed the responses to the first four statements. We looked at the respondent's position and project duration (shown in Figures 7a and 7b) following questions raised by the audience when we presented preliminary results from this study at the 27th NASA/IEEE Software Engineering Workshop.[15]

We can see that the managerial and technical staff generally had very similar perceptions

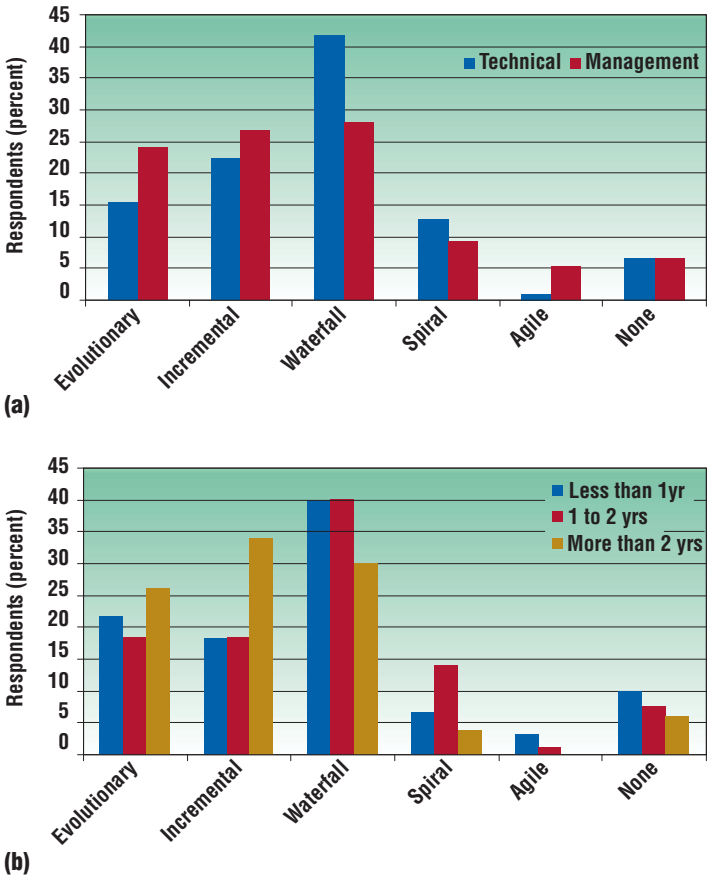


Figure 3. Lifecycle model used split by (a) position (*management* represents executives, architects, managers, and consultants; *technical* represents technical specialists, analysts, and designers) and (b) project duration.

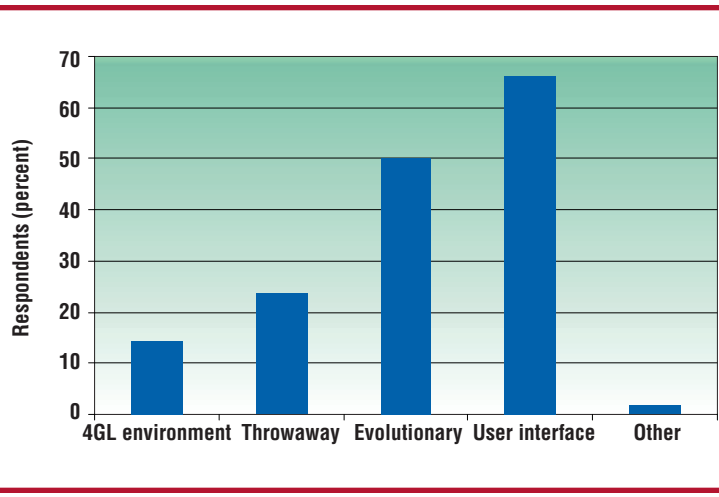


Figure 4. Prototyping type.

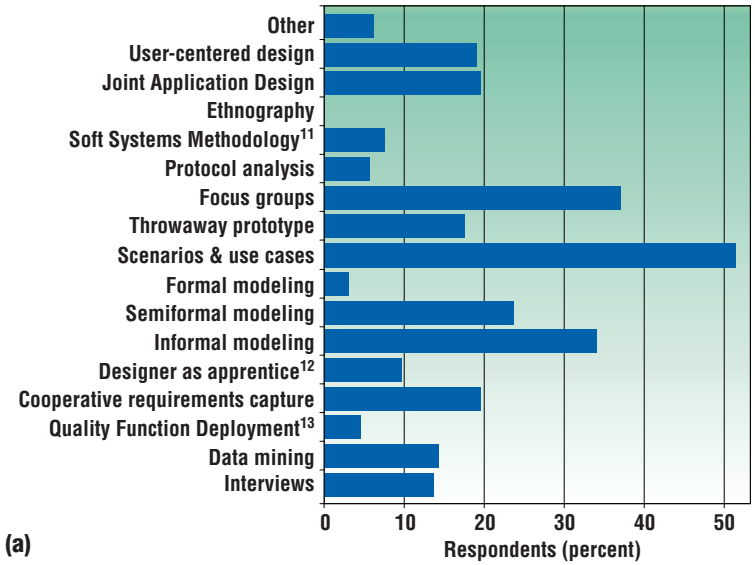


Figure 5. Techniques used for requirements (a) elicitation and (b) modeling.

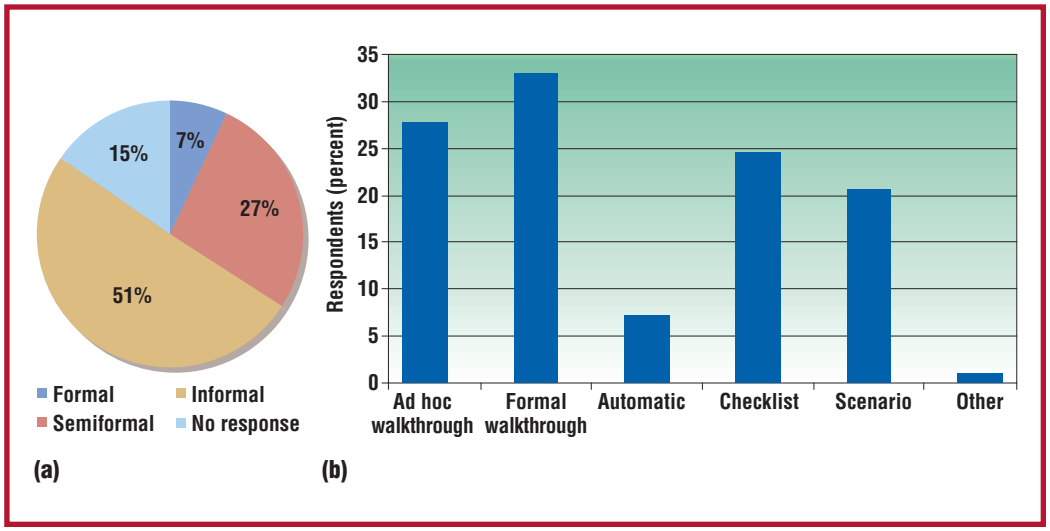
regarding the end products' ease of use and suitability to customers. They had reasonably similar perceptions regarding the completion of projects on time and within budget, although the technical staff seemed slightly less positive about these. "Perceptions about project budget, duration, and end-product quality varied somewhat from conventional wisdom. For example, only around 40 percent of respondents thought that projects were late or over budget."

When we examine the responses with respect to the project duration, however, we see considerably less parity. Indeed, when projects run longer than two years, managerial and technical staff believe that they keep to schedule and budget only about 20 percent of the time, compared to 60 percent of projects lasting up to one year. Responses for projects lasting beyond two years were, on the whole, more negative.

We've provided real data regarding the practices employed in the software industry as a reference point rather than opining on those practices. Nonetheless, we'd like to identify some of the more notable findings, which we've phrased as statements. We hope these will instigate further study by ourselves and other researchers:

- Formal models are rarely used.
- Ad hoc development practices do not impact end-product quality.
- The waterfall lifecycle model is still popular.
- OO techniques are not dominant.

Figure 6. Requirements (a) formality of modeling notation and (b) inspection techniques.



- Industry perception is that we are not failing as much as many suspect, at least in the shorter duration projects.

We'd like to continue collecting data and invite interested readers to visit the survey site (www.personal.psu.edu/cjn6/survey.html) and participate.

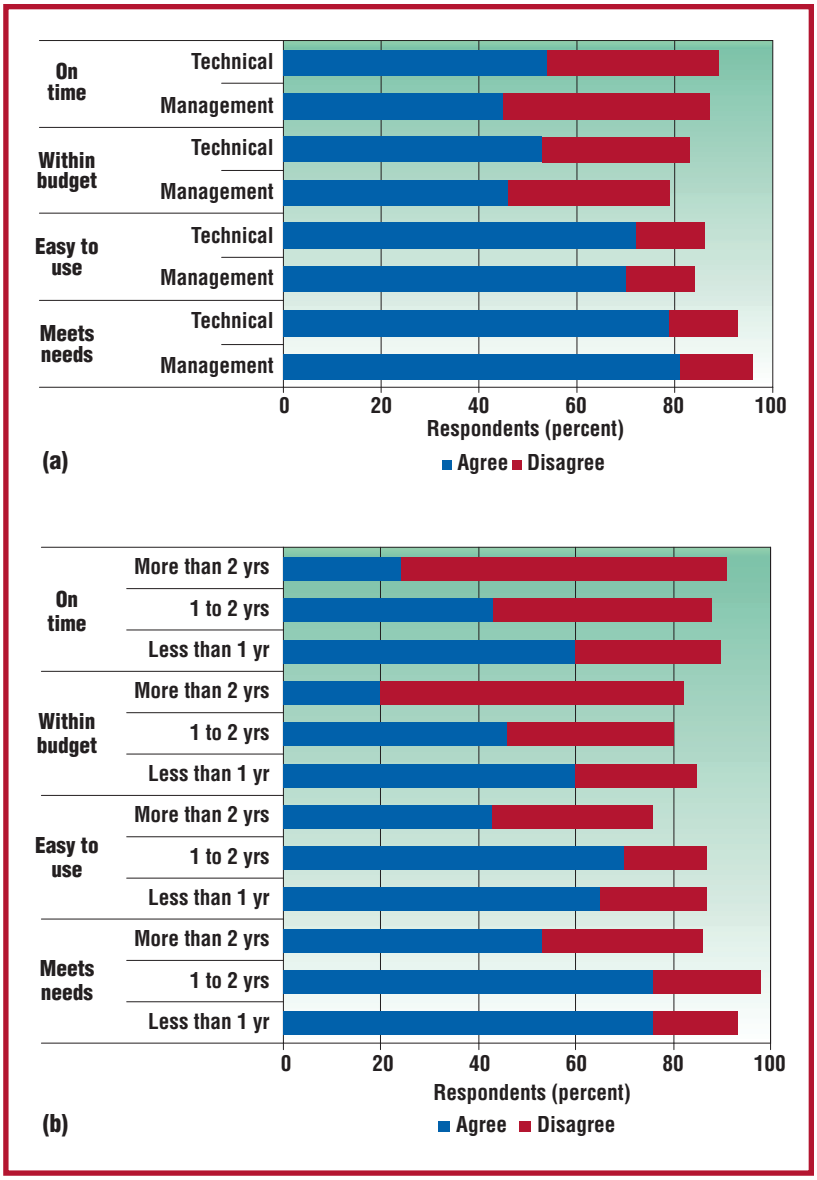


Figure 7. Proportion of agreement or disagreement with four statements regarding project success split by (a) position and (b) project duration.

References

1. B. Curtis, H. Krasner, and N. Iscoe, "A Field Study of the Software Design Process for Large Systems," *Comm. ACM*, vol. 31, no. 11, 1988, pp. 1268–1287.
2. K. El Emam and N.H. Madhavji, "A Field Study of Requirements Engineering Practices in Information Systems Development," *Proc. 2nd IEEE Int'l Symp. Requirements Eng.*, IEEE Press, 1995, pp. 68–80.
3. U. Nikula, J. Sajeniemi, and H. Kalvianen, *A State-of-the-Practice Survey on Requirements Eng. in Small- and Medium-Sized Enterprises*, tech. report, Telecom Business Research Ctr., Lappeenrata Univ. of Technology, 2000.
4. D. Zowghi, D. Damian, and R. Offen, "Field Studies of Requirements Engineering in a Multi-Site Software Development Organization: Research in Progress," *Proc. Australian Workshop on Requirements Eng.*, Univ. of New South Wales, 2001; www.cs.uvic.ca/~danielad/AWRE/Zowghi_AWRE.pdf.
5. C. McPhee and A. Eberlein, "Requirements Engineering Questionnaire," http://sern.ucalgary.ca/~cmcphee/RE.
6. C. McPhee and A. Eberlein, "Requirements Engineering for Time-to-Market Projects," *Proc. 9th IEEE Int'l Conf. and Workshop Eng. of Computer-Based Systems* (ECBS 02), IEEE CS Press, 2002, pp. 17–24.
7 A. Cockburn, *Writing Effective Use Cases*, Addison-Wesley, 2000.
8. J.M. Carroll, "Encountering Others: Reciprocal Openings in Participatory Design and User-Centered Design," *Human-Computer Interaction*, vol. 11, no. 3., July 1996, pp. 285–290.
9. J.H. August, *Joint Application Design: The Group Session Approach to System Design*, Yourdon Press, 1991.
10. L.A. Macaulay, *Requirements Engineering*, Springer-Verlag, 1996.
11. P.B. Checkland and J. Scholes, *Soft Systems Methodology in Action*, John Wiley & Sons, 1990.
12. H.R. Beyer and K. Holtzblatt, "Apprenticing with the Customer," *Comm. ACM*, vol. 38, no. 5, May 1995, pp. 45–53.
13. Y. Akao, *Quality Function Deployment*, Productivity Press, 1990.
14. I. Sommerville, *Software Engineering*, 6th ed., Addison-Wesley, 2001.
15. P.A. Laplante, C.J. Neill, and C. Jacobs, "Software Requirements Practices: Some Real Data," *Proc. 27th NASA/IEEE Software Eng. Workshop*, IEEE CS Press, 2002, pp. 121–128.

About the Authors

Colin J. Neill is the professor in charge of software engineering at the Pennsylvania State University's Great Valley Graduate Center and codirector of the Software Engineering Group. His research interests encompass all aspects of software engineering, from requirements engineering to software auditing. He received his PhD in software engineering from the University of Wales Swansea. He is a member of the IEE and IEEE. Contact him at Eng. Div., Penn State Great Valley, 30 E. Swedesford Rd., Malvern, PA 19355; cjn6@psu.edu.

Phillip A. Laplante is an associate professor of software engineering at Penn State Great Valley and codirector of the Software Engineering Group. His research interests include real-time and embedded systems, image processing, and software requirements engineering. He received his PhD in computer science from Stevens Institute of Technology. He cofounded the *Real-Time Imaging* journal and edits two book series, including the CRC Press series on image processing. He is a member of the International Society for Optical Engineering and ACM and a senior member of the IEEE. He is also a licensed professional engineer in Pennsylvania. Contact him at Eng. Div., Penn State Great Valley, 30 E. Swedesford Rd., Malvern, PA 19355; plaplante@psu.edu.