Software Engineering Institute

# Quantifying Uncertainty in Expert Judgment: Initial Results

Dennis R. Goldenson
Robert W. Stoddard II

**March 2013**

**TECHNICAL REPORT**
CMU/SEI-2013-TR-001
ESC-TR-2013-001

**Software Engineering Measurement and Analysis**

http://www.sei.cmu.edu

Carnegie Mellon

This report was prepared for the

SEI Administrative Agent
AFLCMC/PZE
20 Schilling Circle, Bldg 1305, 3rd floor
Hanscom AFB, MA 01731-2125



DM-0000137

# Table of Contents

# List of Figures

# List of Tables

# Acknowledgments

First of all we offer our heartfelt thanks to the individuals who participated in the training exercises and provided the data from which we drew our results. We clearly would have nothing to report without them. Our colleagues in the SEI's cost estimation research group were instrumental in myriad ways to the design and implementation of this study as well as the larger project on Quantifying Uncertainty in Early Lifecycle Cost Estimation (QUELCE) of which this work is a part. They include in particular Bob Ferguson, Jim McCurley, Dave Zubrow, and Mike Zuccher from the SEI, Eduardo Miranda of the Carnegie Mellon School of Computer Science, and Ricardo Valerdi of the University of Arizona.

Eduardo Miranda also was instrumental in recruiting study participants from the Carnegie Mellon School of Computer Science. Ricardo Valerdi coordinated access to defense and industry participants in a master class held in Melbourne, Australia. R. Krishnan, Susan Timko, and Andy Wasser kindly helped us recruit study participants from Carnegie Mellon's Heinz College. Teri Reiche, the director of Carnegie Mellon's institutional review board, was extremely helpful to us in obtaining appropriate authorization for our research involving human subjects. Jill Diorio and Jerry Pottmeyer of the SEI's ethics compliance team were similarly helpful in gaining access to sites for conducting the training on the Carnegie Mellon campus as well as in their usual responsibilities for work of this kind.

Doug Hubbard graciously gave us permission to reuse some of his large catalog of generic test questions. Mike Cullen and his engineering leadership team at Tektronix provided valuable insights for implementing and refining our methods during an early site visit there. The SEI's Julie Cohen, John Foreman, and Tim Morrow participated in another early calibration training session where they also recognized the importance of such exercises to quantify expert judgment for senior DoD personnel responsible for risk analysis and program management. In addition to John Foreman, others from the SEI and the DoD who have been particularly helpful at various stages in this research include Rob Flowe, John Robert, Mike Philips, David Scherb, and Maryann Watson.

As always we owe a great debt of gratitude to Erin Harper who is a valued colleague on the research as well as an outstanding technical writer. Finally we very much appreciate the leadership of Anita Carleton, Michael May, Bill Scherlis, and Doug Schmidt in the SEI and DoD management chain for their recognition of the importance of this research and approval of funding for it.

# Abstract

The work described in this report, part of a larger SEI research effort on Quantifying Uncertainty in Early Lifecycle Cost Estimation (QUELCE), aims to develop and validate methods for calibrating expert judgment. Reliable expert judgment is crucial across the program acquisition lifecycle for cost estimation, and perhaps most critically for tasks related to risk analysis and program management. This research is based on three field studies that compare and validate training techniques aimed at improving the participants' skills to enable more realistic judgments commensurate with their knowledge.

Most of the study participants completed three batteries of software engineering domain-specific test questions. Some participants completed four batteries of questions about a variety of general knowledge topics for purposes of comparison. Results from both sets of questions showed improvement in the participants' recognition of their true uncertainty. The domain-specific training was accompanied by notable improvements in the relative accuracy of the participants' answers when more contextual information to the questions was given along with "reference points" about similar software systems. Moreover, the additional contextual information in the domain-specific training helped the participants improve the accuracy of their judgments while also reducing their uncertainty in making those judgments.

# 1 Introduction

## 1.1 The Research Problem

Reliance on expert judgment is indispensable for unprecedented Major Defense Acquisition Programs (MDAPs). Many decisions, both technical and managerial, rely on expert judgment in the absence of sufficient historical data throughout the system lifecycle. Yet research and experience show that experts often are overconfident and overoptimistic in their judgments under uncertain conditions [4, 7, 9, 11, 14]. A major part of our larger research agenda on Quantifying Uncertainty in Early Lifecycle Cost Estimation (QUELCE), the work described here aims to develop and validate methods for calibrating expert judgment in early DoD cost estimation. Reliable expert judgment is crucial for many tasks across the program acquisition lifecycle, including cost estimation and perhaps most critically for tasks related to risk analysis and program management.

There is a large literature on overconfidence and optimism in expert judgment. While the literature on the effectiveness of training to calibrate the judgment of individual experts is smaller, results show that calibration training can lead to marked improvements in the trainees' judgment skills [12]. However, the literature focuses most on reducing over-confidence. More work is necessary to better understand how to make judgment more realistic and accurate. Moreover, much of the literature relies on generic questions about a wide variety of general knowledge even though the expertise needed is usually domain-specific [12].

The QUELCE method relies on expert judgment at several steps, including (a) the identification of "change drivers" that can affect the costs of a given project over its lifecycle, (b) the identification of states within a change driver, (c) the probability of a change driver departing from a nominal status, (d) the strength of the cause-effect relationship between one change driver and another, and (e) identifying any significant interactions between change drivers that may jointly affect a third change driver. The outputs of the QUELCE method serve as inputs to existing cost estimation models [19, 6]. Expert judgment must be consistently dependable and repeatable to be credible within cost estimation. Therefore, a method is needed to ensure that expert judgment is satisfactorily calibrated before experts participate in the QUELCE method.

## 1.2 Research Method

Our research includes a series of field studies to compare and validate training techniques aimed at improving expert judgment skills. Our current focus is on training to improve individual judgment skills to enable participants to make more realistic judgments commensurate with the state of their knowledge [13].

We followed a phased approach to reduce risk before doing experiments with the DoD or contractor personnel. As seen in Section 3, participants in the first three studies were Carnegie Mellon University software engineering graduate students, members of the SEI technical staff, and participants in a cost estimation master class in Australia. We used this initial phase to refine our understanding and use of domain-specific questions and "reference points" before undertaking more expensive and logistically difficult experiments with defense experts.

As shown in more detail in Section 2.3, and Appendix A, our domain-specific questions provide much more contextual background information than do the typical, short ”trivial pursuit” questions that have been used in other research on calibrating judgment or risk literacy. We chose to ask questions about existing software systems in the first phase of our research. The reference points provide comparable information about systems from similar application domains. While there is increasing recognition in the research that expertise is domain specific [4, 12], to our knowledge such questions and reference points are unprecedented in the research literature on calibrating expert judgment.

Our training materials emphasize the importance of recognizing uncertainty. However, the ultimate goal also includes improving the accuracy of estimates for use in QUELCE and other decision making under uncertain conditions. Discussion and information sharing about various heuristics was meant to help the participants establish reasonable bounds of uncertainty around their answers to the test questions. The increased contextual information in the domain-specific questions and reference points was meant to narrow those bounds around the correct answers as the participants considered other pertinent factors.

In addition to recording their answers to the test questions, the participants completed a short feedback questionnaire at the end of the domain-specific training (see Appendix C). We used paper forms to collect the data in the first two field studies, but we replaced the paper with custom software support in the third study.

As shown in Appendices A, and C, the software keeps the study participants from making a number of errors typically made when completing paper forms. A major time saver for both the study participants and us, the software also relieves study staff from misinterpreting undecipherable handwriting.[1]

[1] As shown in Appendix B, the reference points still are limited to paper. Resources permitting, we may modify the current software user interface for use in distance learning to allow participants to see only a single question at a time with no back referencing to compare with their earlier answers. We also intend to improve the interface for querying the reference points more flexibly and efficiently.

# 2 Experimental Design

## 2.1 Participant Guidance and Training

The individual calibration training began with a brief introduction that included guidance about how to make more realistic judgments tempered with a degree of confidence to reflect the participants' actual knowledge. Experts often are expected to know the "right" answer. We stress that it is vital to recognize what remains uncertain under as yet unknown circumstances.

The introductions were followed by a series of three or four calibration exercises. Each exercise started with a battery of factual questions. The questions asked the trainees to provide upper and lower bounds within which they were 90 percent certain the correct answer was included. Each test battery was followed immediately by a brief review of the correct answers. A short discussion followed where the students were given further guidance about ways to explicitly consider interdependencies among related factors that might affect the basis of their best judgments under uncertain circumstances.

The guidance included heuristics about the following:

- ways to increase the odds of being right
- thinking of other factors beyond the questions themselves that might affect the pros and cons of being right
- adjusting answers based on previous feedback
- avoiding "anchoring" on an initial "best" answer before thinking about why you may be wrong
- thinking first about why you might be wrong and then reducing your uncertainty based on your knowledge of related things

We limited the size of each training session to a maximum of 15 participants to make the training more manageable to conduct and valuable for the participants. The sessions were small enough to encourage wide-ranging discussion and active learning among the participants. Small sessions also allowed us to incrementally increase the total number of cases and diversity of the total sample. Each training session took 2½ to 3 hours. As shown in Section 3 we included both generic and domain-specific questions to test hypotheses about the comparative effectiveness of their use in the training.

## 2.2 Selection of Study Participants

Participation in the study was entirely voluntary in compliance with approval for research on human subjects by the Carnegie Mellon Institutional Review Board. Since most of the training exercises were held outside of normal class time we relied on flyers, email, and the good offices of faculty colleagues and deans to encourage participation. Light meals and snacks were provided as appropriate for the time of day. Participants received a report of their own performance and the overall results. Anything that could identify the participants personally was of course held in strict confidence and stored separately in a secure manner with access limited to the research team.

## 2.3 Data Collection and Management

Four generic calibration test batteries were used that included 20 short questions each. Examples from the first battery are in Table 1.

*Table 1: Example Generic Test Questions*

| |
|---|
| How many feet tall is the Hoover dam? |
| What percentage of aluminum is recycled in the US? |
| In 1913, the US military owned how many airplanes? |
| The first European printing press was invented in what year? |
| In what year was Harvard founded? |
| What is the wingspan (in feet) of a Boeing 747 jumbo jet? |

It required more time to consider the domain-specific questions since they and the associated reference points included more contextual information than the generic questions. Hence the domain-specific test batteries were limited to 10 questions in each of three batteries. Examples from the first battery are in Table 2. The contextual information for each software system in the left column was followed by the question itself in the right column. The full question set is shown in Appendix A.[2]

*Table 2: Example Questions from Domain-Specific Test Battery 1*

| | |
|---|---|
| **Epiphany** is the web browser for the GNOME desktop. GNOME (GNU Network Object Model Environment) runs on Unix-like operating systems, most notably Linux. Powered by the WebKit engine, Epiphany aims to provide an uncomplicated user interface that enables users to focus on Web content instead of the browser application. | How much total effort in person years has been spent on this project? |
| **Apache JAMES Project:** A complete and portable enterprise mail engine based on open protocols; also a mail application platform that allows processing emails, e.g., to generate automatic replies, update databases, filter spam, or build message archives. | What is the project's current codebase size in LOC? |
| **LibreOffice:** A multi-platform, integrated office suite based on copyleft licenses and compatible with most document formats and standards: Includes spreadsheet, word processor, chart, business productivity, presentation, database, linix, C++ and other applications. | How much total effort in person years has been spent on this project? |
| **OpenGroupware.org** is a set of applications for contact, appointment, project, and content management. It is comparable to Exchange and SharePoint portal servers. It is accessible using Web interfaces and various native clients, including Outlook. Its servers run on almost any GNU/Linux system, can synchronize with Palm PDAs, and are completely scriptable using XML-RPC. | What is the current codebase size in LOC? |

The first domain-specific test battery included only a limited amount of contextual information about the software system about which each question asked. These questions were meant to get the study participants thinking about what they needed to consider in making realistic judgments under uncertain conditions, while recognizing the need for more contextual information to make well-informed judgments.

[2] We crafted the software engineering domain-specific questions and reference points from information available from Ohloh (www.ohloh.net/). Ohloh is a directory that provides links to many project source code repositories and provides "factoid" metrics for thousands of open source projects.

We provided the study participants with additional information about the questions in the second and third domain-specific test batteries. We also introduced at that time the use of the reference points to provide comparable information about similar software systems. Examples from the second battery are shown in Table 3. The contextual information for these batteries and reference points was limited to other factors, the knowledge of which might help the study participants answer the questions. Additional questions asked about the projects in the second and third domain-specific batteries included the same ones asked in the first battery: (1) "What is the project's current codebase size in LOC?" and (2) "How much total effort in person years has been spent on this project?" The full question sets for the second and third domain-specific test batteries are shown in Appendix A. The reference points can be found in Appendix B. The format differs, but the reference points contain the same kinds of contextual information used in domain-specific test batteries 2 and 3.[3]

*Table 3: Example Questions from Domain-Specific Test Battery 2*

| | |
|---|---|
| **Mercurial** is a fast, lightweight Source Control Management system designed for efficient handling of very large distributed projects.<br>---<br>Over the past twelve months, 130 developers contributed new code. This is one of the largest open-source teams in the world, and is in the top 2% of all project teams in our database. Over the entire history of the project, 458 developers have contributed. The first lines of source code were added in 2005.<br>---<br>LOC = 152,551 \| 14% comment to code ratio \| 39 person years of effort | What percentage of the code is written in the product's major language (Perl)? |
| **Google Chrome:** The open-source project behind Google Chrome (Chromium) builds on components from other open source software projects, including WebKit and Mozilla: It is aimed at improving stability, speed and security with a simple and efficient user interface.<br>---<br>Established codebase: The first lines of source code were added in 2008. The project has seen a substantial increase in activity over the last twelve months.<br>---<br>C++ = 39%; C = 33%; XML = 8%; HTML = 6%; Other =14%<br>LOC = 5,535,674 \| 1683 person years of effort | What is the ratio (%) of comments to LOC in the current codebase? |
| **Mozilla Calendar** project develops Mozilla Sunbird (a stand-alone calendar application) and Lightning, a calendaring extension for Mozilla Thunderbird. Their goal is to bring Mozilla-style ease-of-use to your calendar, without tying you to a particular storage solution.<br>---<br>Over the past twelve months, 157 developers contributed new code to Mozilla Calendar. This is one of the largest open-source teams in the world, and is in the top 2% of all project teams in our database. Over the entire history of the project, 495 developers have contributed. The first lines of source code were added in .<br>---<br>C++ = 32%; JavaScript = 29%; XML = 15%; C = 7%; CSS = 7%; Java = 5%; Other = 5%<br>LOC = 927,266 32% comment to code ratio 253 person years of effort | In what year were the first lines of source code added? |

[3] The correct answers to the questions that the study participants had not yet answered remained hidden from view in the reference points as well as the questions.

# 3 Experimental Results

As shown in Table 4, a total of 36 individuals participated in the study during FY 2012. The first of three separate groups consisted of Carnegie Mellon University software engineering graduate students along with a few members of the SEI technical staff. The second group consisted of members of a master class led by Ricardo Valerdi and Dave Zubrow in conjunction with the Improving Systems and Software Engineering Conference (ISSEC) held in Melbourne, Australia in August 2012. The third group consisted of Carnegie Mellon University graduate students from the Heinz College along with two additional software engineering graduate students. All of the graduate students had previous industrial experience. We kept the three groups small to encourage active learning and class discussion.

*Table 4: Number of Study Participants*

| Venue | Number of Test Cases | | |
|---|---|---|---|
| | Total | Domain-specific | Generic |
| 1: Carnegie Mellon Graduate Students and Software Engineering Institute Technical Staff Members | 21 | 14 | 14 |
| 2: Australian Master Class Participants | 8 | 8 | 0 |
| 3: Carnegie Mellon Graduate Students | 7 | 7 | 0 |
| Totals = | 36 | 29 | 14 |

A total of 29 individuals from all three groups completed three batteries of software engineering domain-specific test batteries. A total of 14 participants from the first group also completed four batteries of generic knowledge questions that often are used for training meant to calibrate recognition of uncertainty.

As noted in Section 2 our domain-specific questions included much more contextual information than the generic knowledge questions. We also provided reference points to give the study participants additional information about software systems similar to the ones in the questions.

The results for both sets of questions showed improvement over the test batteries that were consistent with studies in other domains with respect to recognition of the participants' true uncertainty. The domain-specific training was accompanied by notable improvements in the relative accuracy of the participants' answers when we introduced the additional contextual information to the questions and the reference points about similar software systems.

## 3.1 Calibrating Judgment

### 3.1.1 Characterizing Uncertainty

A simple summation of the number of times that the correct answer for a calibration test questions falls within the upper and lower bounds specified by the study participants is commonly used to measure calibration of expert judgment. Such a measure characterizes the idea of recognizing people's uncertainty reasonably well. In fact, faculty with whom we have collaborated in these studies who teach software engineering graduate courses in cost estimation have used their stu-

dents' uncertainty in answering questions similar to our domain-specific software engineering questions (but without the additional contextual information) as a teaching moment to get the students thinking about what else they need to know to inform their technical skills and manage their time and attendant risks. Our results are consistent with prior research using the same measure that we call "accuracy-within-bounds." That is true particularly for our domain-specific questions.

We used box plots to summarize the distributions of the study participants' scores over this measure and two other derived measure. Box plots as originally envisaged by Tukey [21] make no assumption of statistical normality. They are simply based on distribution of the data by percentiles. As shown in Figure 1, the box runs from the first through the third quartile (25th and 75th percentiles) of the entire data distribution. The distance between the two ends of the box is called the interquartile range; it contains half of the observations (study participants in this report). The whiskers, which may exist both above and below the box, extend to the outermost data points within another 1½ times the interquartile range. Asterisks above or below the whiskers are classified as outliers (i.e., cases that are unusually large or small).

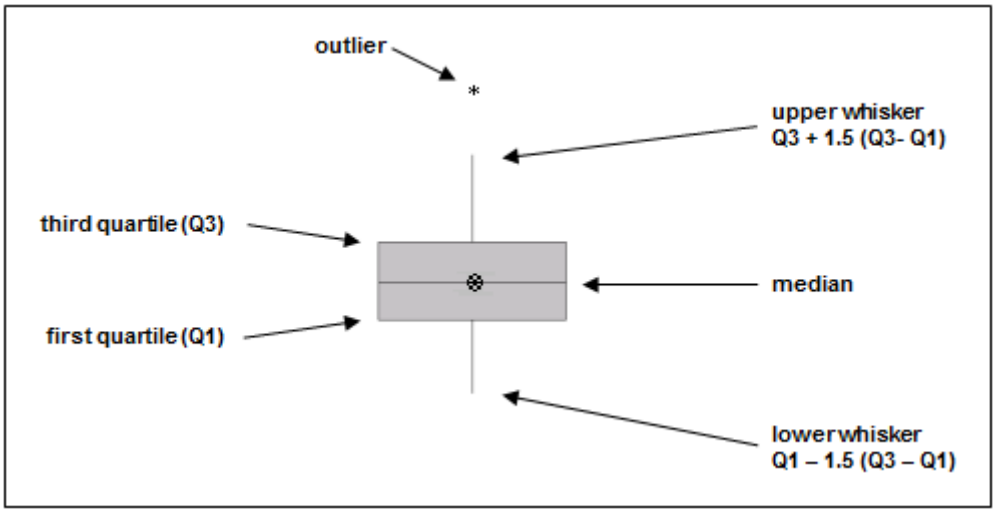


*Figure 1: Interpreting Box Plots*

As shown in Figure 2, the median proportion of study participants whose upper and lower bounds included the correct answers to the generic questions rose from 40 percent in the first test battery to 80 percent in the fourth battery ($p < .0002$).[4] Notably, the participants' median proportions rose from 10 percent to 70 percent over the course of only three domain-specific test batteries ($p < .00001$).

[4] All of the significance tests for the box plots throughout this report are based on the Mann-Whitney U-test. The U-test can be used determine the probability that the medians of two distributions are significantly different. However, with larger samples, it also can detect important differences in the shape and spread of the distributions. A succinct description of the U-test can be found in *The SAGE Encyclopedia of Social Science Research Methods* [1].

The slight dip in generic test battery 3 is statistically insignificant. As shown in Figure 4 on page 10, however, the inconsistency for a different measure of accuracy in test battery 3 is much larger. That is because the measure of accuracy-within-bounds confounds accuracy with precision.

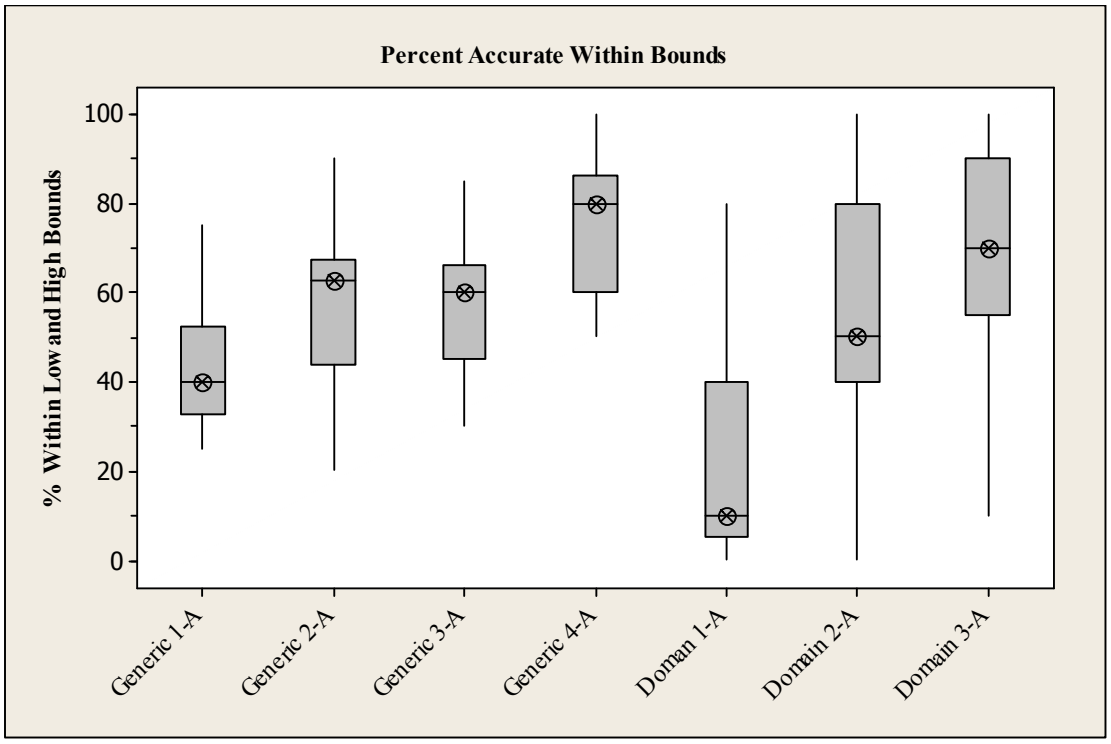


*Figure 2: Accuracy-Within-Bounds by Test Battery*

While the measure of accuracy-within-bounds is a reasonable way to characterize the participants' recognition of uncertainty, it is an imperfect measure of accuracy in making judgments under uncertain conditions. Participants in calibration training exercises sometimes improve their chances of being accurate by unrealistically expanding the width of their confidence bounds to recognize their true uncertainty. However, it is not enough to know how often study participants or true domain experts are able to establish confidence bounds that contain the correct answers on calibration tests. It is equally important to achieve sufficient confidence in those answers.

Recognizing one's uncertainty is a major lesson of calibration training. Students are encouraged to think about what else they already know that can inform their judgments beyond what is stated explicitly in the test questions. Other unstated factors can be equally or more important. However, narrowing the distance between one's lower and upper confidence limits is also a major goal of calibration training. That is why we included reference points in our domain-specific training exercises.

Such an approach is even more important in real-world practice situations involving requirements analysis, portfolio management, performance modeling, risk analysis, program management, and

many other competences in addition to cost estimation. That is why we are building a working prototype repository of DoD domain-specific information in FY 2013.[5]

### 3.1.2 Our Focus on Precision and Accuracy

Both precision and relative accuracy are extremely important. Experts need to narrow their bounds of uncertainty and consider how close their best estimates can come to what proves to be accurate over the passage of time. Yet this is not emphasized widely in the existing literature on calibration of expert judgment. As illustrated in Figure 3, for example, having a more precise expert judgment that is closer to being correct is preferable even if the bounds around the expert's best judgment do not include the correct answer.

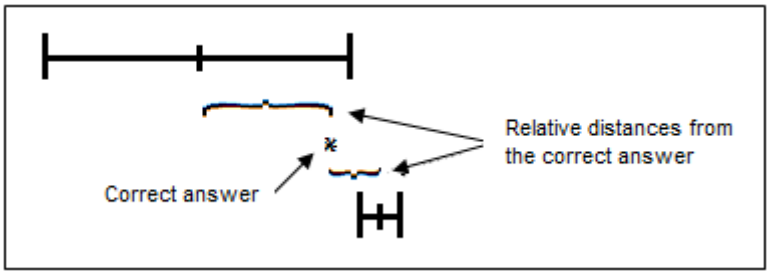


*Figure 3: Balancing Accuracy and Precision*

We used two independently defined derived measures to characterize the study participants' relative accuracy and precision.[6] Our measure of relative accuracy simply takes the absolute value of the distance between the participant's best judgment and the accurate score for each test question. It then divides that value by the accurate score for the question to normalize for differences between the units of measure among the questions in the full test battery. The derived measure for each participant is the median score over all of his or her answers in that test battery.

$$median \sum_{i=1}^{n} \frac{ABS\ (BestJudgment - AccurateScore)}{AccurateScore}$$

Where

$$BestJudgment = median\ (LowerBound \ldots UpperBound)$$

A score of zero would indicate that all of the study participant's answers on the test battery were exactly correct. Hence higher median scores are worse than lower ones.

As shown in Figure 4, the pattern of test scores for study participants who were trained with generic examples shows steady improvement over time with the exception of test battery 3, which has the worst median test score for the participants as a group in the four test series (0.51). The

[5] Development of this Software Cost Analysis Repository (SCAR) is a major activity for the larger project on Quantifying Uncertainty in Early Lifecycle Cost Estimation (QUELCE).

[6] We also are experimenting with creating a single measure that summarizes both precision and closeness of best estimates to exact scores without confounding the two.

bump-up from battery 2 to battery 3 is a significant one ($p < .03$). The greater difference here as compared to Figure 2 is not surprising for a measure that expresses accuracy independent of precision and in relative rather than absolute terms.

Still, with the exception of battery 3, the overall pattern of improvement for the generic test questions improved over time from medians of .46 in test battery 1 to .24 in test battery 4 ($p < .0004$). However, the pattern of improvement between test battery 1 and battery 3 for the domain-specific examples was both consistent and markedly greater over time. There the medians for the participants as a group improved from a much higher .88 to .19 in the third battery ($p < .00001$). We think that is because the study participants were chosen for their familiarity with software engineering and because the domain-specific contextual information in the questions and reference points provided the participants with a realistic basis for considering their answers.

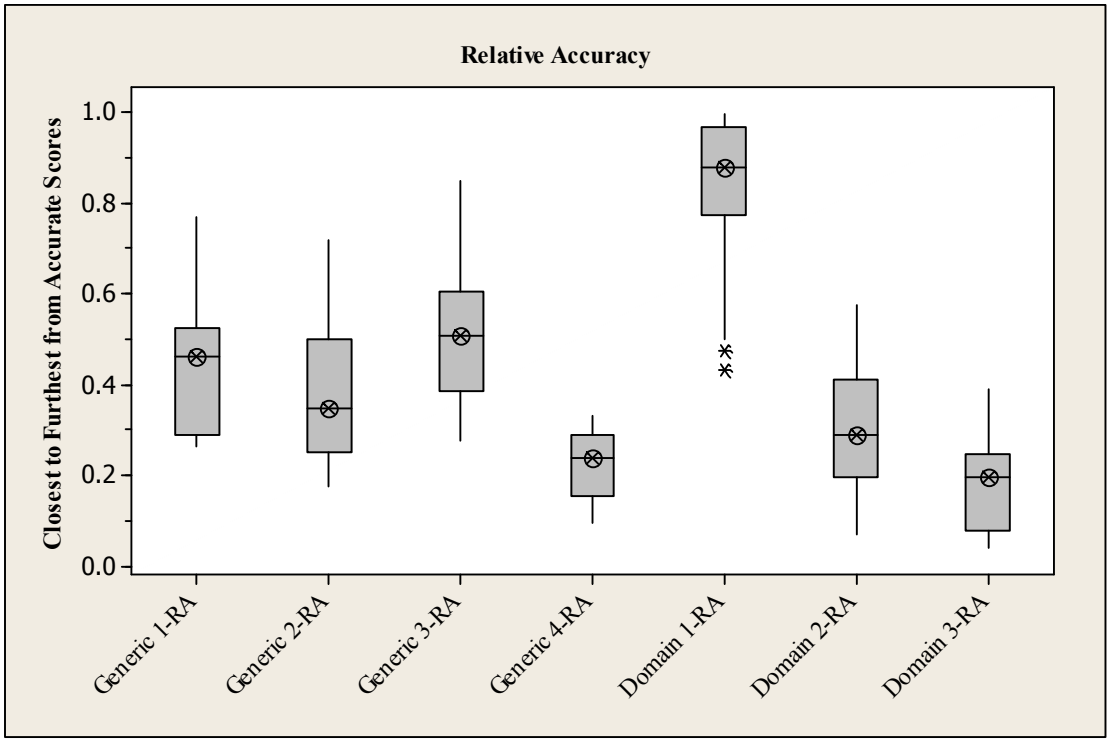


*Figure 4: Relative Accuracy by Test Battery*

Similarly our measure of precision takes the absolute value of the distance between a participant's high and low bounds of certainty for each test question. It, too, divides that value by the accurate score for the question to normalize for differences between the units of measure among the questions in the full test battery. The derived measure for each participant is the median score over all of his or her answers in that test battery.

$$median \sum_{i=1}^{n} \frac{ABS\ (LowerBound \ldots UpperBound)}{AccurateScore}$$

Higher median scores are worse than lower ones. A score of zero would indicate that all of that participant's high and low bounds on the test battery were exactly alike.

No consistent pattern of change in the precision scores was apparent in Figure 5. As expected for both the generic and domain-specific training and their respective test batteries, the study participants tended to widen their confidence bounds over time to realistically express their uncertainty in answering the test questions. The medians for the generic test batteries rose from .50 to .83 ($p < .07$).

The differences were more pronounced for the domain-specific tests, where the study participants started with somewhat narrower confidence bounds than those who participated in the generic training. Their median scores as a group rose from .31 to .48 over the three test batteries ($p < .17$). However, the domain-specific participants also widened their confidence bounds more initially, from .31 in battery 1 to .70 in battery 2 ($p < .01$).

Notice, too, though that the participants also tended to narrow their confidence bounds in answering the questions in the third domain-specific test battery, with a drop in median scores from .70 to .48 ($p < .03$). We remain cautious in interpreting this last finding at this stage in our research, but it does suggest that training aimed at improving expert judgment under uncertain conditions can begin to improve realistic confidence along with accurate judgments of fact.

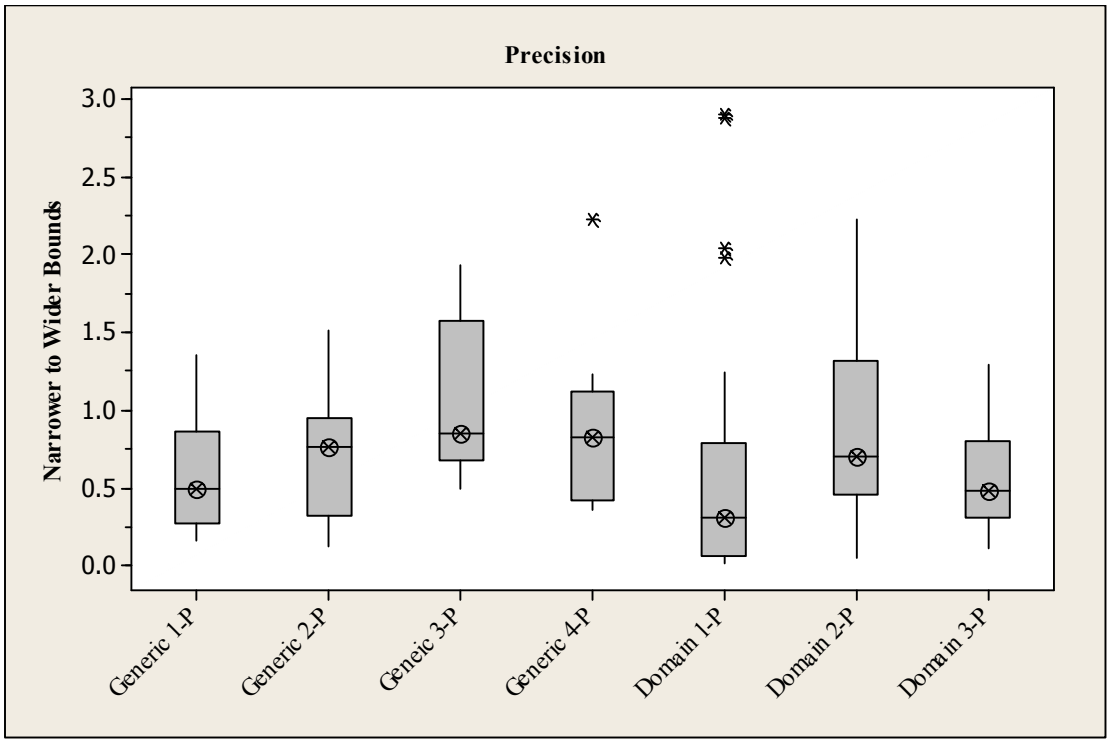


*Figure 5: Precision by Test Battery*

### 3.1.3 Summarizing Relationships Among the Three Measures

The relationships between relative accuracy and accuracy-within-bounds for the generic and domain-specific test batteries are summarized in Figure 6 and Figure 7 respectively. As expected,

the patterns of relationships for most of them are relatively weak[7] and statistically insignificant.[8] Relative accuracy and accuracy-within-bounds appear to be measuring two different things.

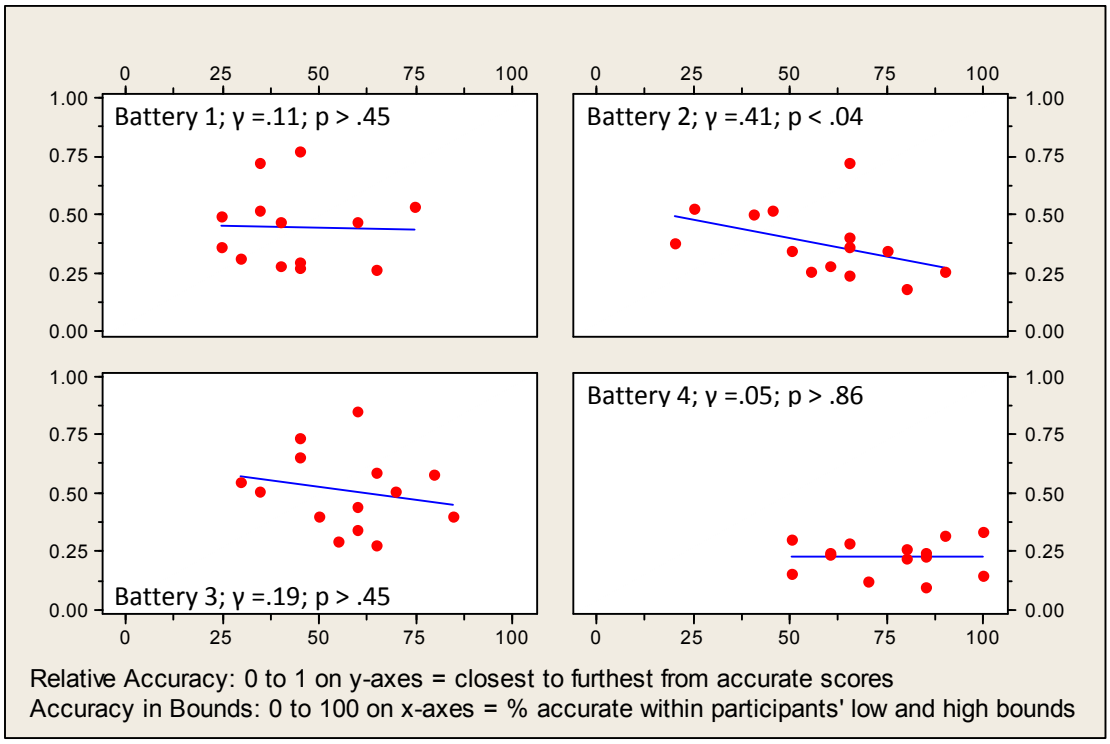


*Figure 6: Relative Accuracy by Accuracy-Within-Bounds: Generic Test Batteries*

[7] Goodman and Kruskal's gamma (γ) is an ordinal measure of association that is appropriate for both categorical and poorly distributed numerical data [8]. A proportional reduction in error (PRE) statistic with an intuitive interpretation, the value of gamma is the proportion of paired comparisons where knowing the rank order on one variable reduces the proportionate error in predicting the rank order on the other variable. So, for example, if gamma is .75 then knowing the rank order of the observations on that variable reduces our error in predicting the ranks of the other variable by 75 percent.

[8] Standard statistical tests can be misleading for data such as these. There is a good deal of noise in poorly distributed data such as these when treating slight differences in numeric scores ordinally. The p-values also can be very much affected by outliers and small numbers of cases. Not surprisingly many of the relationships in the scatter plots in this section are not statistically significant (p > .10).

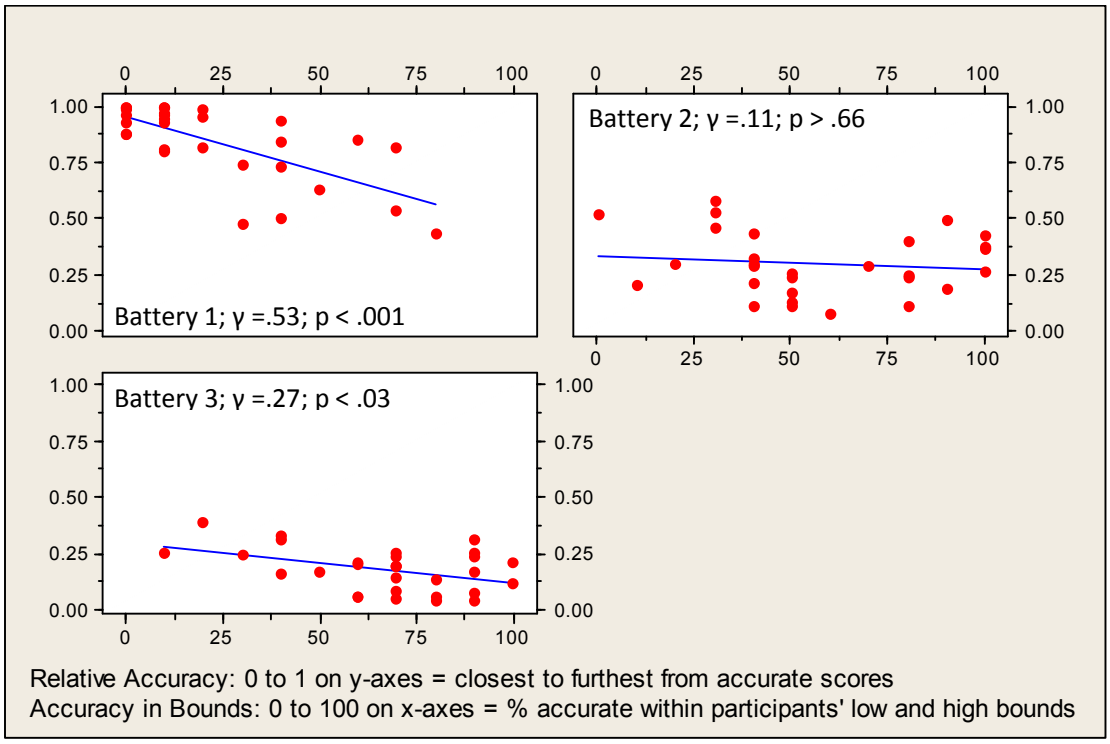


*Figure 7: Relative Accuracy by Accuracy-Within-Bounds: Domain-Specific Test Batteries*

The covariation in Figure 8 and Figure 9 summarizes the extent to which relative accuracy is accompanied by greater precision. They help to address two important questions: "How many of the study participants' best judgments are both relatively accurate and embedded in bounds that are precise enough to provide decision makers with confidence in their judgments?" and "How much calibration training is necessary for experts to realistically recognize the uncertainty in their best estimates?"

As shown in Figure 8, the strengths of the relationships between relative accuracy and precision are weak to moderate at best for all four generic test batteries. The same is so for the three domain-specific test batteries (see Figure 9).

As seen by the clustering of the cases lower on the y-axes of the scatterplots, the study participants' relative accuracy scores improved over both the generic and domain-specific training sessions. As shown in Figure 4, the change is much more pronounced for the domain-specific tests. Similarly, as also shown in Figure 5, the participants' precision scores along the x-axes for the generic test in Figure 8 vary less than those in Figure 9 for the domain-specific test. Again, notice that the precision scores for domain-specific battery 3 cluster much closer to the more precise left side of the x-axis.

Perhaps more importantly, however, the cases cluster most closely in the lower left quadrant of the scatter plot for battery 4 of the generic test questions. The same pattern was even more pronounced by the end of the domain-specific training, even though it included only three test batteries. The study participants as a group became both more accurate and more precise. Almost all of those who participated in the generic training, and all of those in the domain-specific training

groups, can be found in the quadrants closest to the origin of both axes on the two scatterplots. Based on simple sign tests alone, the probability of that occurring by chance is highly unlikely ($p < .0009$ for the generic training and $p < .0001$ for the domain-specific training).[9]

Training aimed at improving expert judgment under uncertain conditions by providing domain-specific contextual information about test questions and information about similar projects does seem to improve realistic confidence along with accurate judgments of fact. Of course the participants in the lower left quartiles of the last test batteries in both Figure 8 and Figure 9 still vary in their accuracy as well as their precision by the end of their calibration training. However, while the best of them (those circled in battery 3 of Figure 9) are somewhat less precise than a few others, their answers are also more accurate. We conjecture that those with the best scores on both dimensions of Figure 9 may be particularly well-suited for making realistic judgments under uncertain conditions.

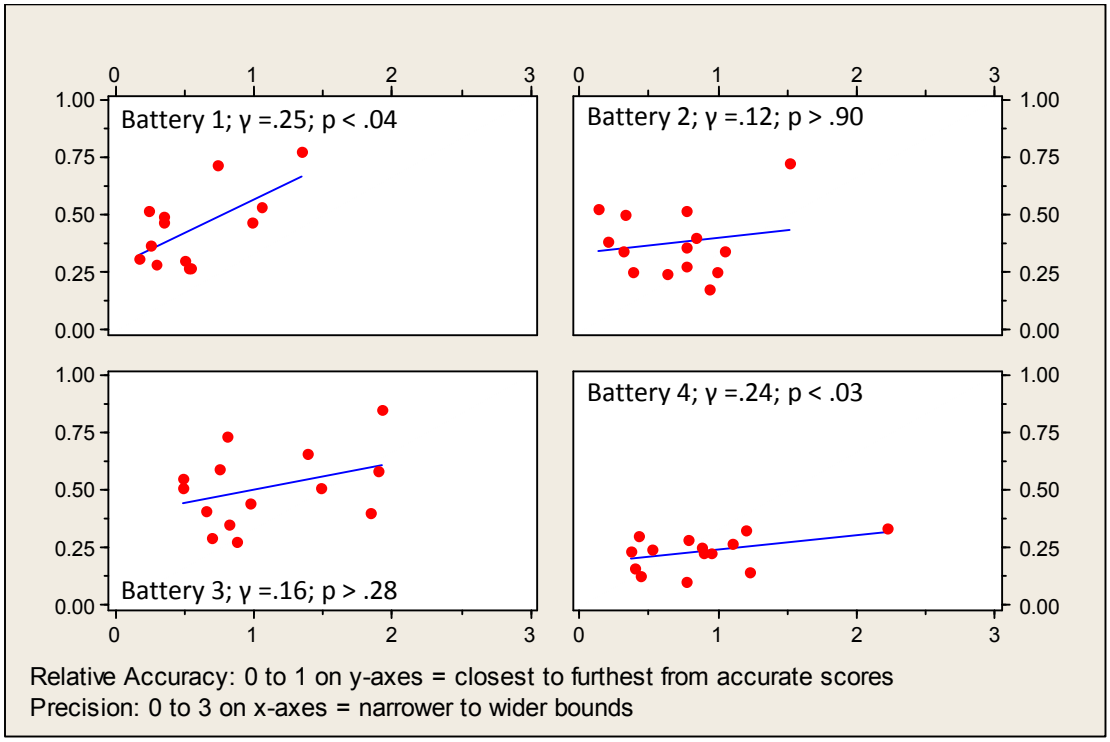


*Figure 8: Relative Accuracy by Precision: Generic Test Batteries*

[9] A basic description of the sign test can be found in Kitchens 2002 [15].

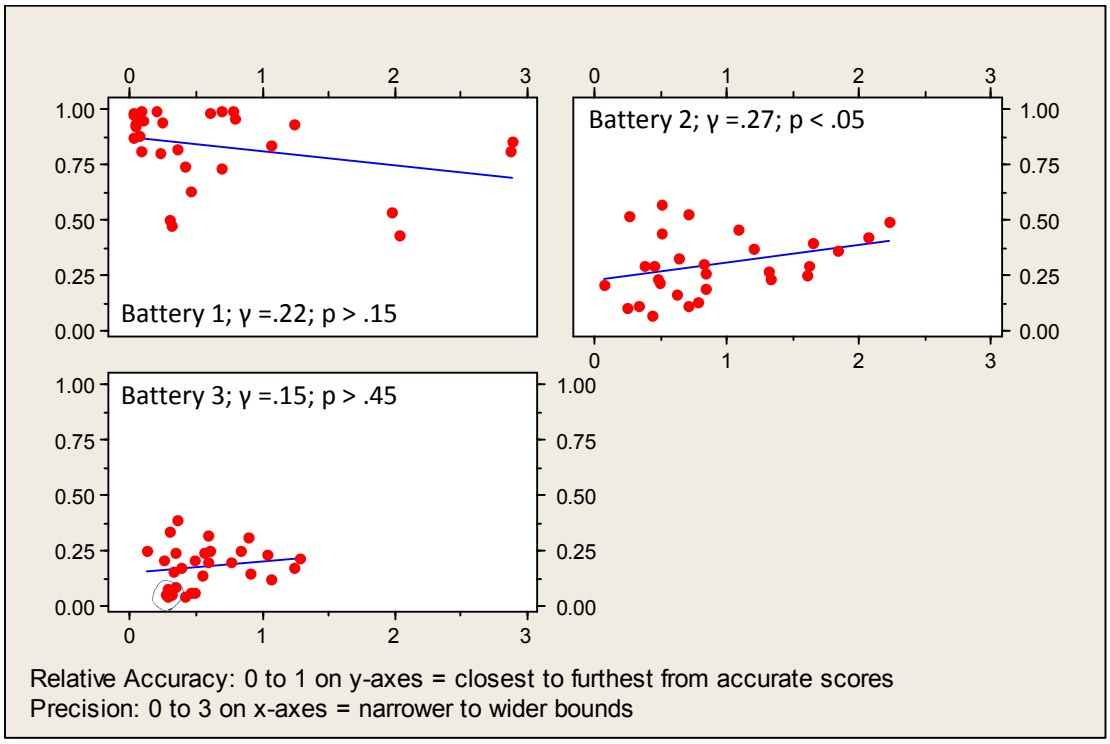


*Figure 9: Relative Accuracy by Precision: Domain-Specific Test Batteries*

Figure 10 and Figure 11 summarize the extent to which accuracy-within-bounds is accompanied by greater precision. Accuracy-within-bounds is a reasonable way to characterize people's recognition of uncertainty. However, not surprisingly, the higher accuracy displayed in the two figures is largely a function of the study participants who have set wider, less precise bounds of uncertainty around their best judgments. That appears to be particularly true for the domain-specific test batteries, especially in battery 2 where the participants who achieved better accuracy-within-bounds scores also are notably less precise than they were in test battery 1. All seven of the relationships in Figure 10 and Figure 11 are quite strong for data of this kind.

Yet, unlike the patterns in the generic interest test batteries, the participants in the third and last of the domain-specific test batteries are all in the left half of the x-axis. Unlike the pattern of relative accuracy on battery 3 in Figure 9, their scores remain distributed widely over the y-axis. However the likelihood of them all being on the more precise side of the x-axis is highly unlikely to have occurred simply by chance ($p < .0001$). This too suggests that providing domain-specific contextual information about test questions and information about similar projects is a valuable way to improve training aimed at improving expert judgment under uncertain conditions.

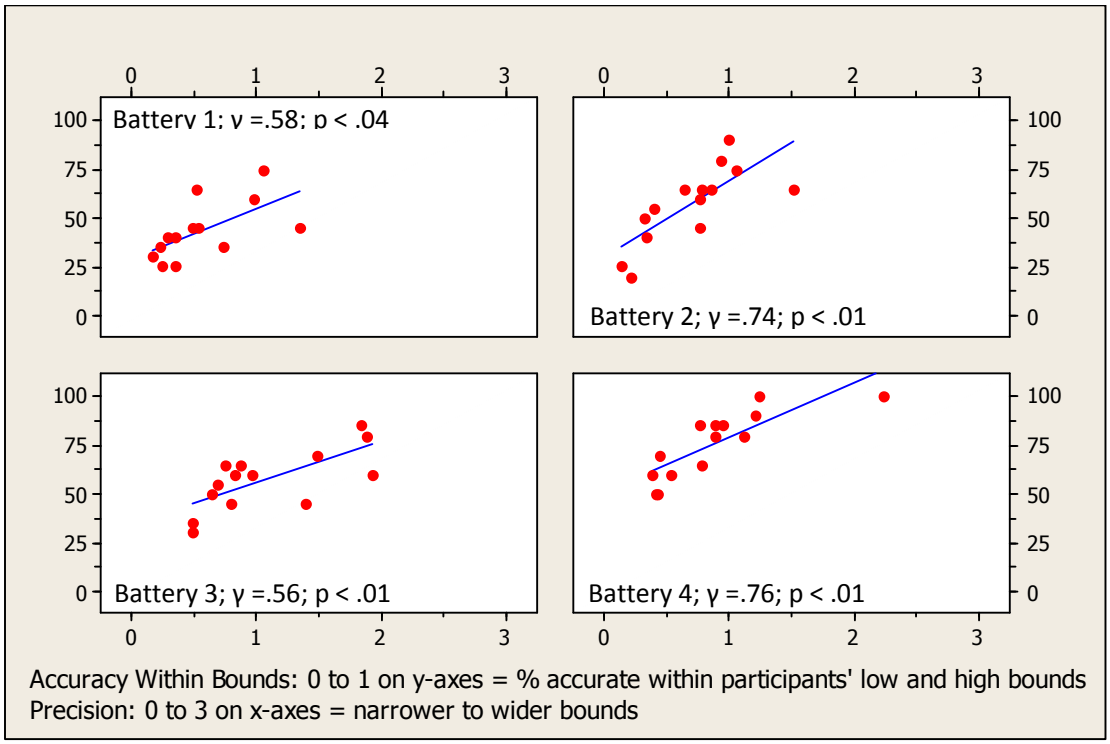


*Figure 10: Accuracy-Within-Bounds by Precision: Generic Test Batteries*

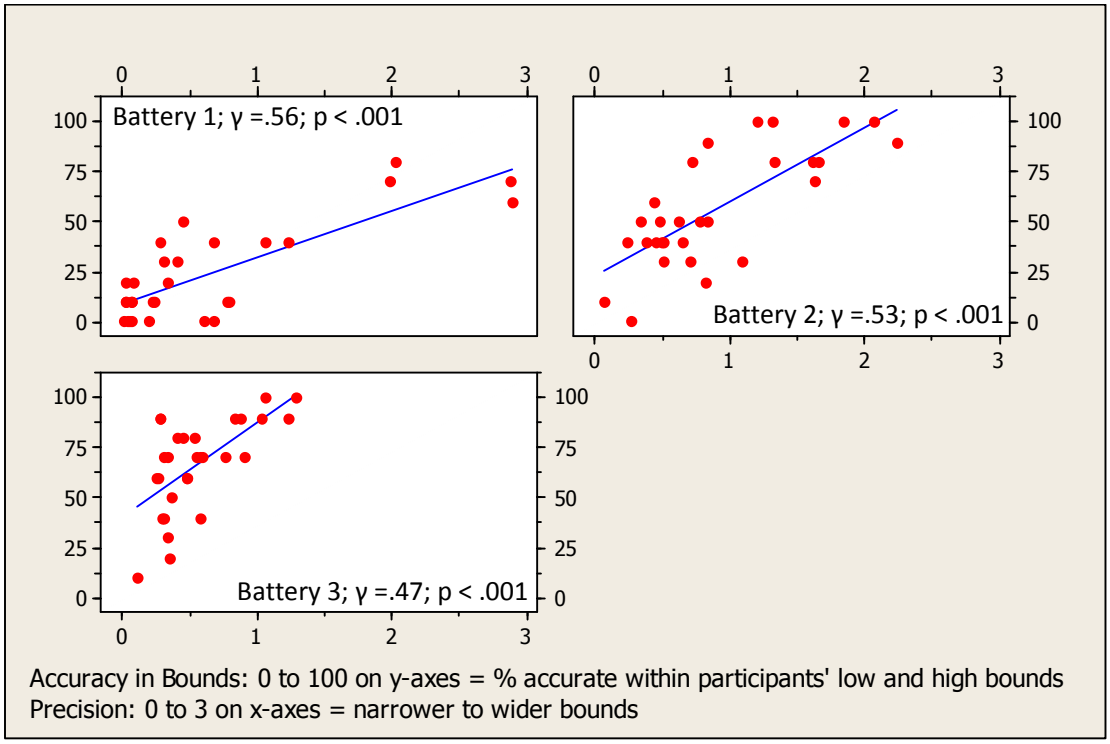


*Figure 11: Accuracy-Within-Bounds by Precision: Domain-Specific Test Batteries*

## 3.2 Impact of Generic Training on Domain-Specific Judgment

Recall from Section 1.2 and earlier in Section 3 that seven of the fourteen participants in the domain-specific training in our first study at Carnegie Mellon University also participated in the generic training session the previous day. This allowed us to compare the performance on the domain-specific tests of those who participated in the generic training with those who did not.

We excluded the participants from the two subsequent replications of the domain-specific training to minimize bias in the results due to differences in the three groups. However, the results in Figure 12, Figure 13, and Figure 14 are consistent with the overall results in Figure 2, Figure 4, and Figure 5 for the entire sample of domain-specific trainees.

While the number of cases is quite small, the differences between the two groups are instructive.[10] Figure 12 displays the differences in the study participants' scores on our measure of accuracy-within-bounds across the three domain-specific test batteries. Those who took the generic training first were always more accurate within bounds than were those who took only the domain-specific training, although the differences narrowed by test battery 3.[11] Those who participated in the generic training before tackling the domain-specific test questions appear to have put bounds around their judgments that more realistically characterize their uncertainty.

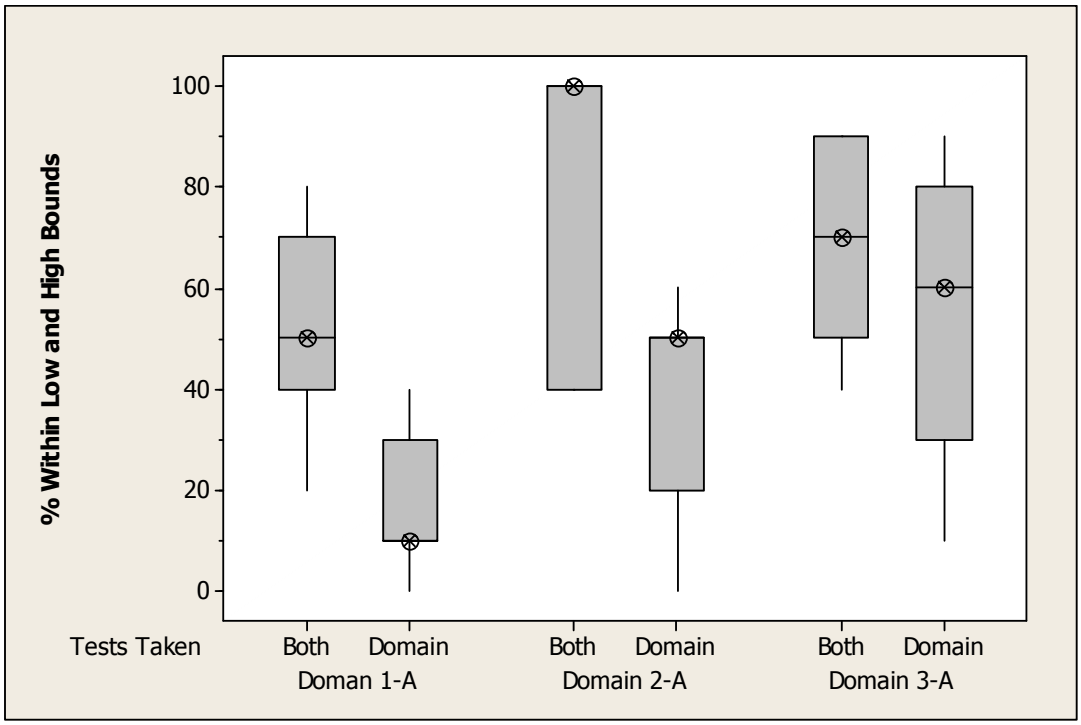


*Figure 12: Accuracy-Within-Bounds by Calibration Training and Domain-Specific Test Battery*

[10] We may do additional studies to increase the number of cases and diversity of the participants.

[11] The differences in the size and shape of the boxes and whiskers cannot be generalized because of the small number of cases. The only statistically significant difference is in battery 2 ($p < .05$).

Figure 13 shows similar results for our derived measure of precision. Those who took only the domain-specific training consistently put narrower bounds around their best judgments than did those who participated in the generic training sessions first. Those who participated first in the generic training put much wider bounds around their best judgments in domain-specific test batteries 1 and 2. The difference in the two groups during battery 2 of the domain-specific training is only marginally significant, however: $p < .01$ and .04 for batteries 1 and 3 respectively.

We cannot know with confidence whether or not the reduction in precision seen for both groups in test battery 3 is realistic without comparing the relationships over time between precision and relative accuracy. Unfortunately the small number of cases does not permit a valid comparison. However, the-full sample scatter plot in Figure 9 suggests that it is commensurate with the participants' improvement in accuracy. Similarly we cannot know whether or not the less precise figures for those who did the generic training first are under-confident. Still it is suggestive that the box plots also narrowed noticeably in test battery 3 for those who took both training sessions.

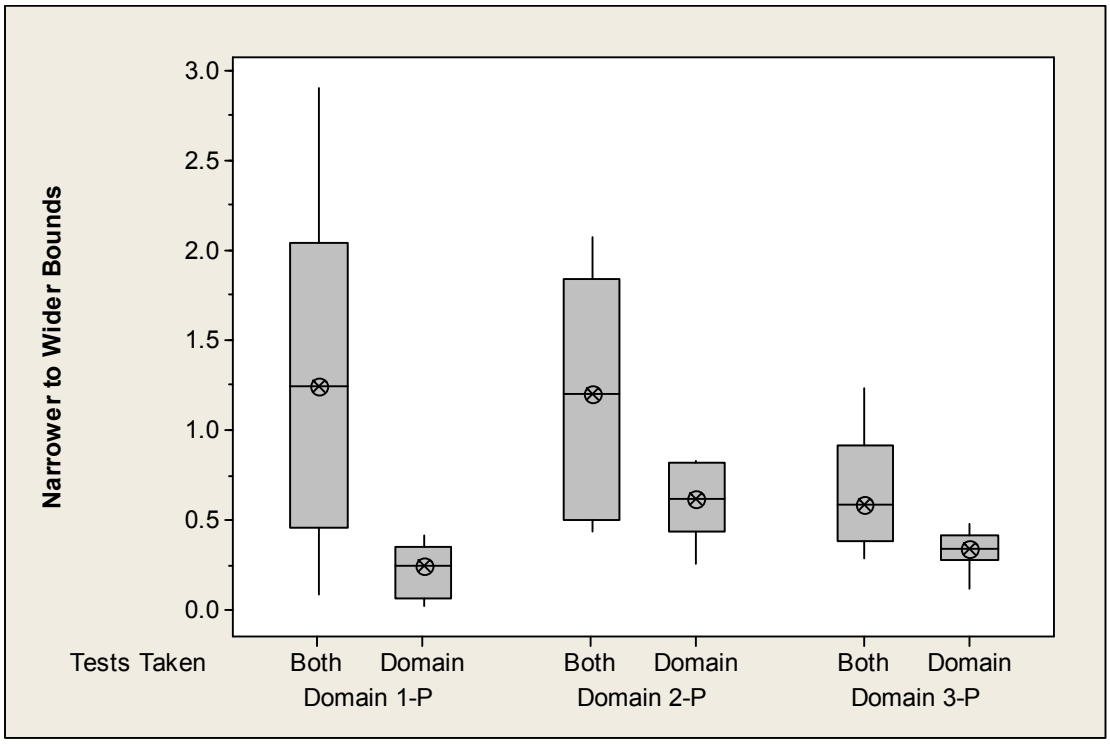


*Figure 13: Precision by Calibration Training and Domain-Specific Test Battery*

A final set of paired box plots is displayed in Figure 14 to summarize the effects on relative accuracy of having done the generic tests first. These plots are much different than those for the effects on accuracy-within-bounds seen in Figure 12. The differences there seem to be because having done the generic tests first encouraged the participants into widening their bounds to calibrate their uncertainty. However, the differences in relative accuracy between those who took the generic training first and those who did not are much less pronounced and none of them are statistically significant.

Moreover, the differences between the two groups are not consistent across the three test batteries. Those who took the generic training appear to have done a bit better in the first and third domain-specific test batteries than those who took only the domain-specific training, although they were somewhat less accurate on test battery 2. However, consistent with the results including the participants from the other two study sites, the major improvement following the introduction of the fuller contextual information in the questions and reference points is evident here in test battery 2.

More research is necessary to better understand the extent to which doing generic training first can affect the results of domain-specific training. Note though in Section 3.3 some of the feedback from participants in our research suggests that doing so may be useful.

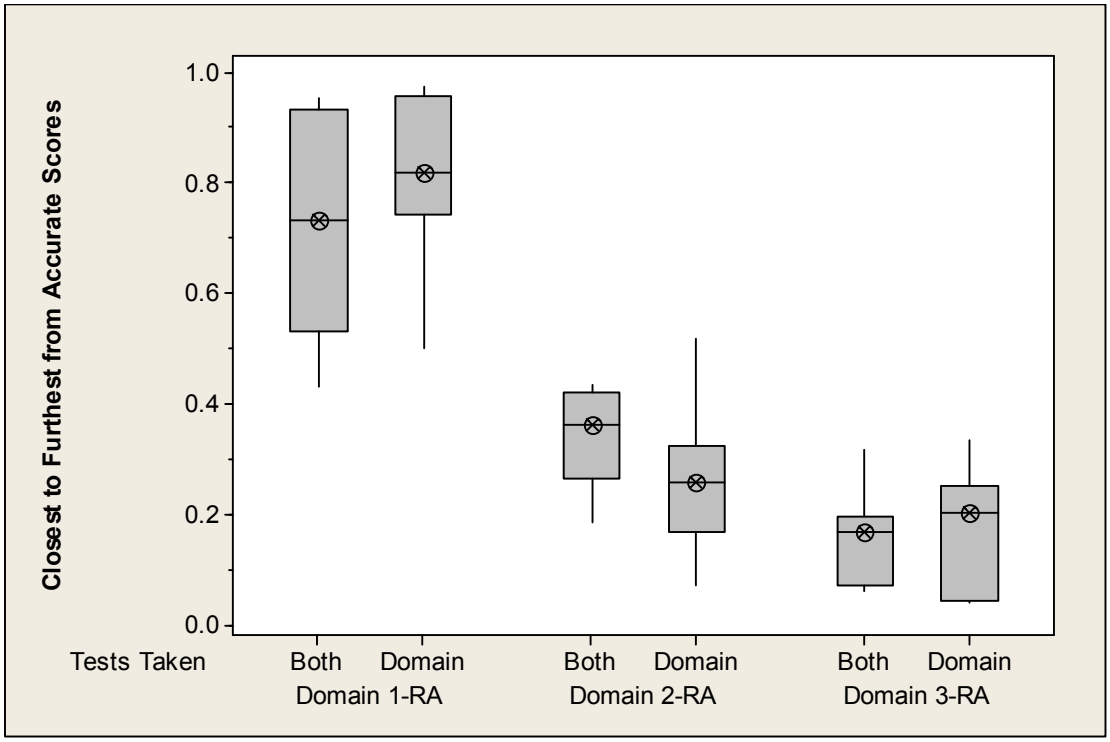


*Figure 14: Relative Accuracy by Calibration Training and Domain-Specific Test Battery*

## 3.3 Participant Feedback

As noted in Section 2.3 the study participants completed a short feedback questionnaire at the end of their domain-specific training (see Appendix C). This section includes a series of 10 figures, each of which summarizes the options the participants chose in answering the 10 feedback questions. We also included several of their selected verbatim responses to four questions to provide a richer sense of the participants' experiences.

The first question asked the participants: "How familiar are you with the kinds of software systems about which we asked today?" The fact that most of them said they had only mixed familiarity with those systems (see Figure 15) suggests that even the limited training we provided in this study has potential for practical use.

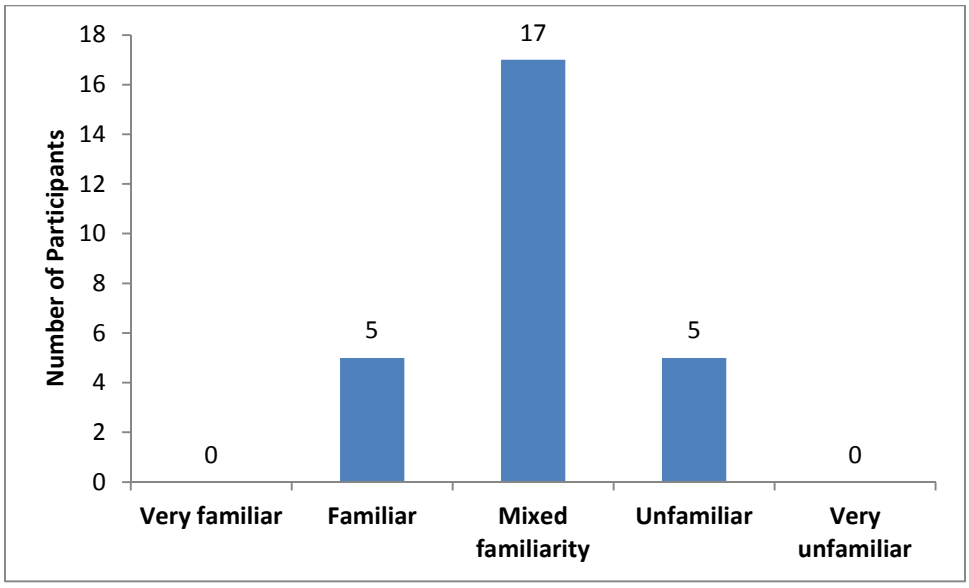


*Figure 15: Familiarity with Software Systems*

One of the main reasons for doing calibration training to improve expert judgment is to help people become aware of the extent of the limitations of making realistic judgments in uncertain circumstances. Hence the next question asked the participants: "Were you surprised about how well or poorly you did?" Only a relative few said they were not surprised (see Figure 16). To avoid asking about two things in the same question, we did not ask the participants in this question whether they did better or worse than expected. That was the reason for question 3, which asked the study participants: "How much difficulty did you have in answering the questions?" As shown in Figure 17, only one study participant said that answering the questions was "reasonably easy." Answering the questions clearly was not a simple task for them.

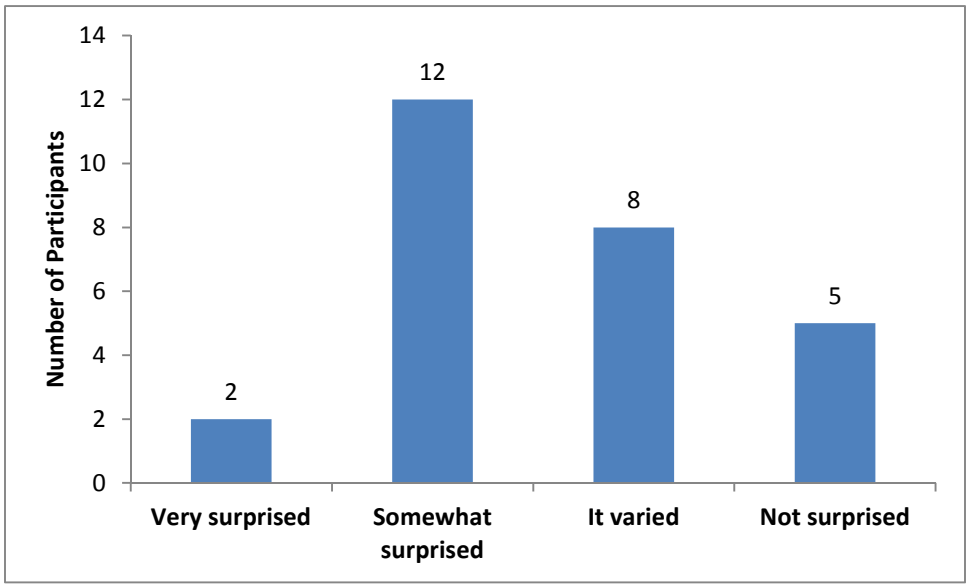


*Figure 16: Participants' Surprise About How They Did*

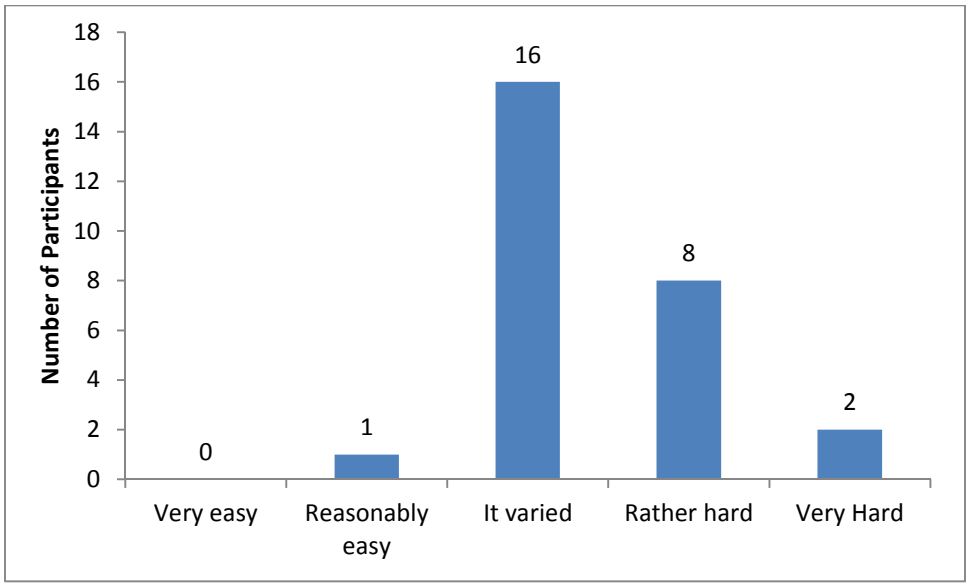


*Figure 17: Difficulty Answering Questions*

We left space after question 3 for the study participants to describe in their own words why they chose their answers from the options that we gave them. Most of them spoke about how hard the task was for them in the first round of testing but emphasized the value added when we provided more contextual information in the questions and domain-specific reference points.

> *Initially I was unaware of many things in giving my answers. I gradually started doing better with more information available.*
>
> *Some of the questions at the beginning seem to be very hard to me even with the given information, but when the test continues, I get more comfortable with estimating the answer.*
>
> *Before. Is reasonably easy after.*
>
> *The data provided was not adequate, especially in the first round.*
>
> *Difficult to estimate date of first line source code.*
>
> *Easy to answer: Hard to get good meaningful answers.*

With the fourth question we asked: "How much guidance would you like to have had today?" As shown in Figure 18 the majority of the study participants would have preferred having more guidance during the training. Possibly for different reasons, several others preferred to leave things as they were. No one preferred having less guidance.

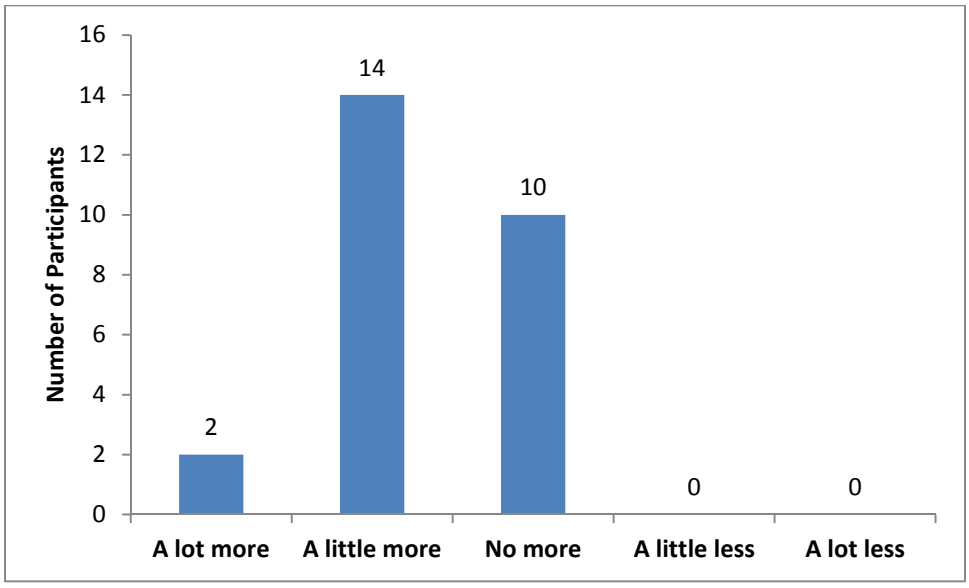


*Figure 18: How Much Guidance Participants Would Have Liked*

In a similar vein, the next question asked: "How much practice would you like to have had today?" While the plurality preferred no change from the existing training, half of the study participants would prefer to have had more practice. Once again, no one asked for less practice (see Figure 19). Recognizing that the number of participants was limited in our studies thus far, their answers bode well as an indicator of the potential for enhancement of such training for use in educational settings and in-service training, including as an integral part of our QUELCE method aimed at quantifying uncertainty in early lifecycle cost estimation.

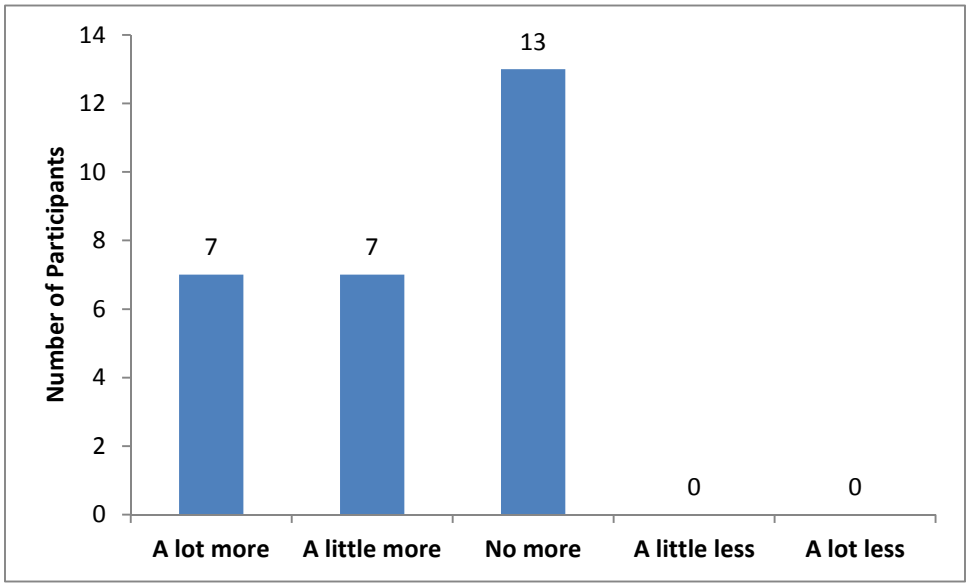


*Figure 19: How Much Practice Participants Would Have Liked*

We used question 6 to ask "Which of the following methods did you use to match your intervals with the state of your knowledge?" The participants chose one or more options, the first four of

which described heuristics that we discussed with them during the calibration training (see Figure 20). The most widely used heuristics involved (1) thinking about other factors that might likely help them make informed judgments in answering the domain-specific questions, and (2) simply widening the intervals between their upper and lower bounds to better recognize their uncertainty about the correct answers. We asked them to describe other ways as well. As shown by the quotes listed immediately after Figure 20, some of their answers described variants on the heuristics we discussed during the training. Note the fourth quotation in particular, which is consistent with our impressions during the training and discussions with some of the study participants after the training.

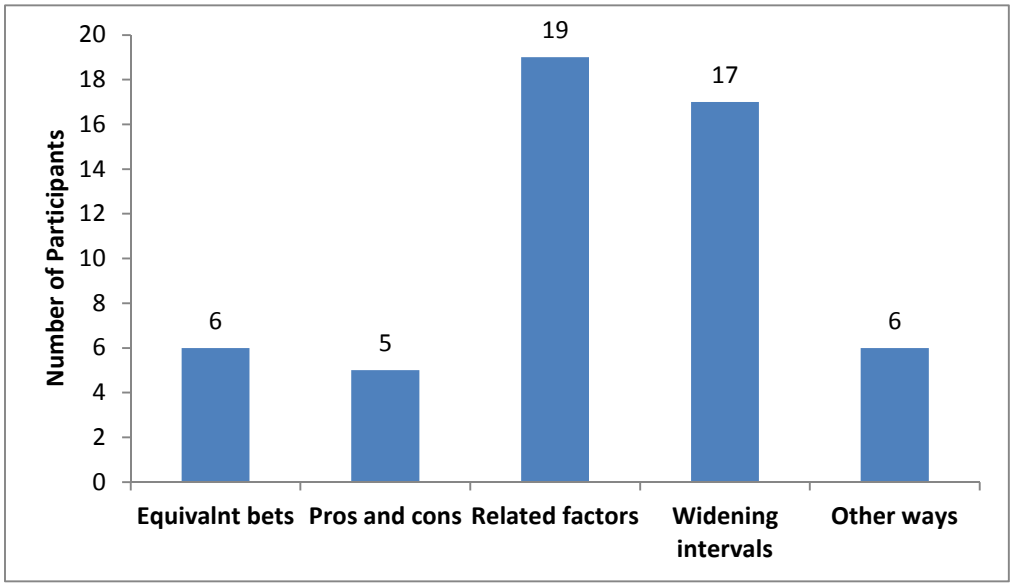


*Figure 20: Methods Participants Used to Match Intervals with State of Knowledge*

> *Picking similar projects that are smaller and larger to set bounds.*
>
> *Looking for parametrics I could use.*
>
> *Solid correlation between some factor.*
>
> *I would try to widen my intervals as much as possible and compared with other software/platform in the same category to accurate my answer.*
>
> *Trying to calculate best/worst scenarios based on info provided.*
>
> *Rules of thumb and ranges based on projects in same domain.*
>
> *Figured rule of thumb for SLOC/Year and took highs + lows as the 90%.*

Questions 7 and 8 asked the study participants about the value of the contextual information in the test questions and reference points: "How informative was the contextual information in the project descriptions shown with the questions?" and "How informative were the tables of 'reference points' describing other projects along with the ones asked about in the questions?" Their answers in Figure 21 and Figure 22 are similar, although the participants found the additional contextual information in the reference points to be somewhat more useful than that in the questions alone.

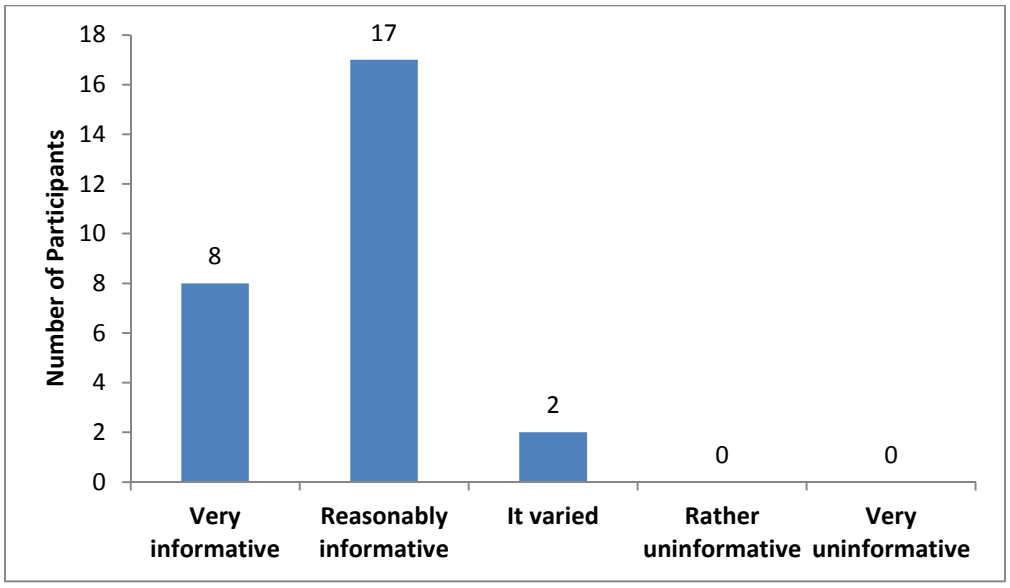


*Figure 21: Informativeness of the Contextual Information in the Test Questions*

> *Good basis: But it was so varied that I had to widen my estimates to feel comfortable,*

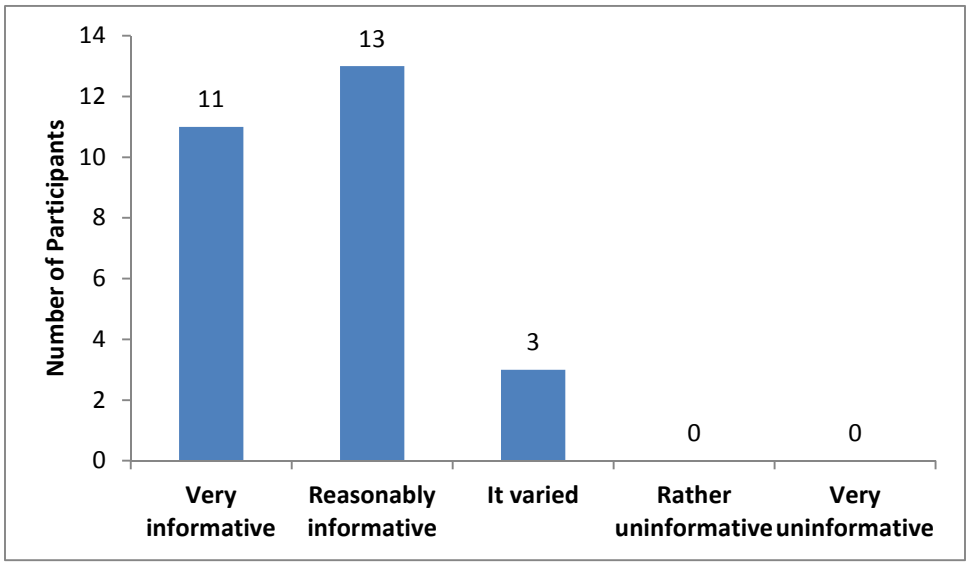


*Figure 22: Informativeness of the Reference Points Tables*

We asked the students one more question on the feedback form about the overall value of the additional contextual information in question 9: "How helpful were the contextual information and reference points?" Once again, as our hypothesis predicted, almost all of those who answered the question found the information quite helpful. Over half of them chose answers that recognized the uncertainty that remained for them in making realistic judgments in answering the questions, and over a third of them found the information to be indispensable (see Figure 23).

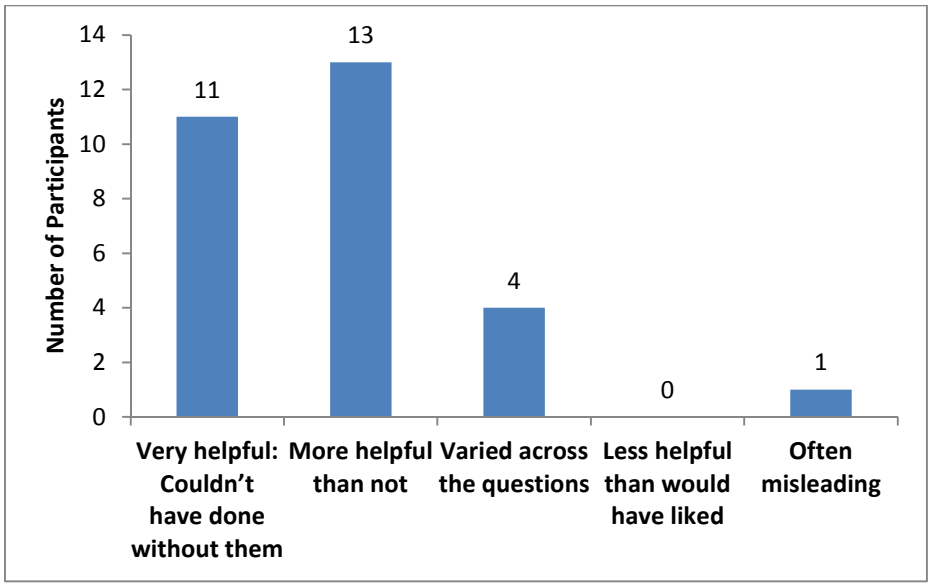


*Figure 23: Helpfulness of the Contextual Information and Reference Points*

The study participants provided some useful insights in response to an open-ended question about other things they considered in answering the domain-specific test questions in their answers to Question 10: "What other kinds of information did you use to inform your decisions?" Notice that some of the participants also used the reference points to cue their thinking about their own previous experience with other software systems.[12]

> *Analogies and relation between available information.*
>
> *Thank you, wonderful, and thought provoking.*
>
> *Normally, person*year : LOC is kind of constant. While some language is verbose in its natural and language like Perl need more comments.*
>
> *Gaming techniques - risks to rules and exploit them.*
>
> *Used the between the values from the tables. Give a margin of error for cases where the context can be slightly different from samples.*
>
> *Some experience, previous questions to set context.*
>
> *Provided refs and any domain knowledge I had. Also, I could remember some rough values from the third set from the second set.*
>
> *Experience of recently used coded ones.*
>
> *Ratios of SOL to py range of 3000 to 4500 per year. Large projects tending 3000/yr small projects tending to 4500/yr.*
>
> *Some past experience and some relation between some of 16 projects (Mozilla-based or Apache ones.*
>
> *My experience: But discounted this as I did poorly on the early tests.*

[12] We've pruned some of the answers to this question in the interest of space.

In Question 11, "What else would you have liked to know?" some of the study participants also provided similarly useful insights in response to another open-ended question about other kinds of information that may have been useful for them in making their judgments. They clearly recognized that additional information often is necessary to make informed decisions under otherwise uncertain circumstances. Some of their answers gave us useful cues for crafting future domain-specific questions and reference points, in particular those that make reference to more detailed information to make better analogies with experience in more closely related situations.[13]

> *Another round to make the ranges tighter.*
>
> *Some low level knowledge ...*
>
> *Length of project phases.*
>
> *More of the maths behind the modeling.*
>
> *Maybe showing some data within time interval would help to estimate the answer with the potential trend of data.*
>
> *A bit more on cost estimation on sustainment side. But it seems you guys are working on that. Overall very useful course. Thank you.*
>
> *Probably more detailed averages and ranges across the domains. Essentially this kind of historical data is very helpful when applying on a context.*
>
> *Development per year to see trends. Developers per year.*
>
> *Relative LOC for languages.*
>
> *Some more better reference points like time size, (current D point), completion of this project.*
>
> *Context about the behavior of a project along the years.*
>
> *How data was extracted from sources (e.g., directly, indirectly via report, or summary).*
>
> *The confidence level of the accuracy of the data.*

Finally, we asked those who participated in both the generic and the domain-specific training about the extent to which their participation in the generic training helped them during the domain-specific training (shown in Figure 24): "If you attended the [generic] session on Monday: How much do you think it helped you think through your answers to the [domain-specific] questions today?"

Only seven people participated in both sessions, and their answers were quite varied. However, one of them entered the following thought-provoking comment on his or her paper feedback questionnaire.

> *Went back to my bad habits for the first test but then widened the range for the last two.*

[13] We have pruned some of the answers to this question in the interest of space.

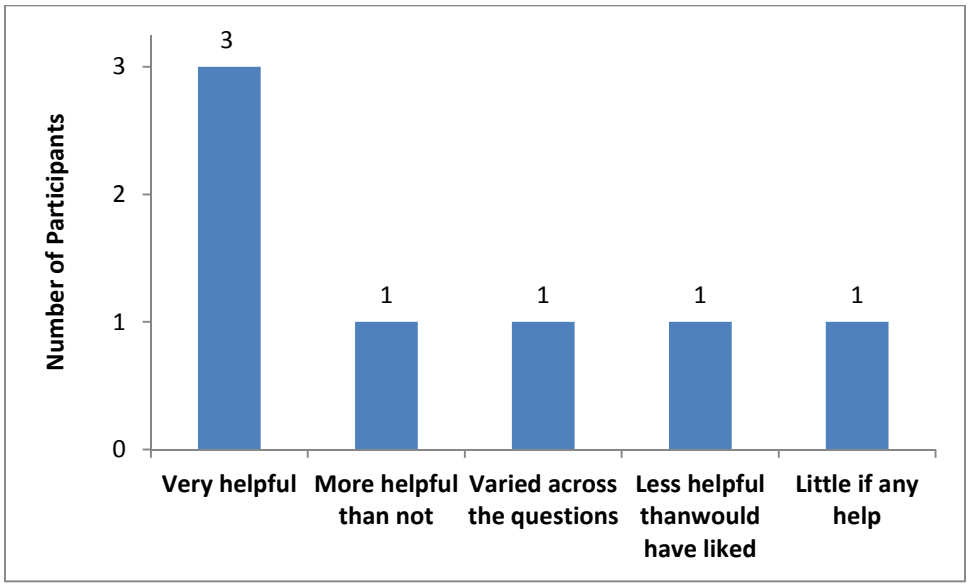


*Figure 24: Value of Generic Training Exercises for Answering Domain-Specific Questions*

# 4 Conclusion

## 4.1 Summary of the Research

A total of 36 individuals from three separate groups participated in this study: Carnegie Mellon graduate students from the School of Computer Science's Master of Software Engineering program and a few members of the SEI technical staff; members of a master class of adult learners in Australia; and graduate students from Carnegie Mellon's Heinz College concentrating on software engineering and information technology along with two more computer science students. All of the participants had previous industrial experience (see especially Sections 1.2 and 2.2).

The calibration training provided guidance about how to make more realistic judgments, tempered with a degree of confidence that reflected the participants' actual knowledge. That guidance was followed by a series of calibration exercises, each of which included a battery of factual questions that asked the trainees to provide upper and lower bounds that they were 90 percent certain included the correct answer to each question. Each test battery was followed immediately by a brief review of the correct answers. A short discussion at the end of the training provided further guidance about ways to explicitly consider interdependencies among related factors that might affect the basis of one's best judgments under uncertain circumstances. We kept the groups small to encourage active learning and class discussion (see especially Sections 2.1 and 2.3).

A total of 29 individuals from all three groups completed three batteries of software engineering domain-specific test batteries. A total of 14 participants from the first study group also completed four batteries of generic knowledge questions (see especially the description of Table 4 early in Section 3).

Results from both sets of questions showed improvement over the test batteries with respect to recognition of the participants' true uncertainty. The domain-specific training was accompanied by notable improvements in the relative accuracy of the participants' answers when we introduced additional contextual information to the questions along with reference points about similar software systems. Moreover, the additional contextual information in the domain-specific questions and reference points helped the participants improve the accuracy of their judgments while also reducing their uncertainty in making those judgments (see Section 3).

## 4.2 Next Steps

Most of the existing research on calibration of expert judgment skills has relied on testing generic knowledge about historical events and physical principles. Our focus will continue to be on testing hypotheses about the value of domain-specific training. Having demonstrated the value of that approach with examples from software engineering, we now are concentrating our energy on DoD domains. We will validate and enhance the existing research by developing DoD domain-specific questions for a series of test batteries associated with the training exercises. We also are investigating the value of providing DoD domain-specific reference points that provide more detailed contextual background about analogous programs as well as the programs being considered in the test questions.

In a related vein, our QUELCE research group currently is working on a project to build a Software Cost Analysis Repository (SCAR) by mining existing DoD data and information repositories. A major part of our FY 2013 research on early lifecycle cost estimation, the intent is to make existing information about MDAPs more widely accessible to DoD personnel through database queries. The repository also will become a useful source of DoD domain-specific questions and reference points for calibration training. In turn, our expert judgment calibration studies will contribute to subsequent studies of the usability and usefulness of the SCAR as well as the incorporation of calibration training as an integral part of the QUELCE method itself.

We are also considering suitable ways to craft succinct DoD domain-specific questions that do not require additional contextual information or reference points. We think that such questions will remain meaningful to more senior DoD and contractor personnel who may not be able to take the time to participate in the kinds of training sessions that we have used thus far. The same or similar questions can be crafted to be appropriate for much less experienced people who otherwise would be overwhelmed with detailed contextual information about specific defense programs or classes of such programs that are unfamiliar to them.

With our colleague Ricardo Valerdi, we are preparing such questions and plan to use them in calendar year 2013 with graduate students at the Air Force Institute of Technology (AFIT). Moreover having succinct DoD domain-specific question sets will enable more realistic experiments and hypothesis tests about the value added by questions and reference points that include additional contextual information.

We also are considering using succinct DoD domain-specific questions with Naval ROTC students at the University of Arizona.[14] It is unlikely that they will be able to improve the accuracy and precision of their answers very much during a brief training session, but recognition of the limitations of judgments made under uncertain circumstances and the need to consider other pertinent factors when making such judgments should be useful learning experiences for them.

In other research with Ricardo Valerdi at the University of Arizona, we will include batteries of true/false questions in our DoD domain-specific studies. Such questions are common in general interest studies of calibration and risk intelligence [4, 13]. Participants answer such questions to the best of their ability and also indicate how confident they are in the accuracy of their answers. Hence a perfectly calibrated individual would correctly answer 90 percent of all the questions in which he or she expressed 90 percent confidence and 60 percent of those for which he or she expressed 60 percent confidence. The individual's over or under confidence can be calculated using a Brier score that was originally created to evaluate the accuracy of meteorologists [2, 23]. A visualization of such a score is in Figure 25.

[14] Such questions may focus initially on operations as opposed to acquisition issues.

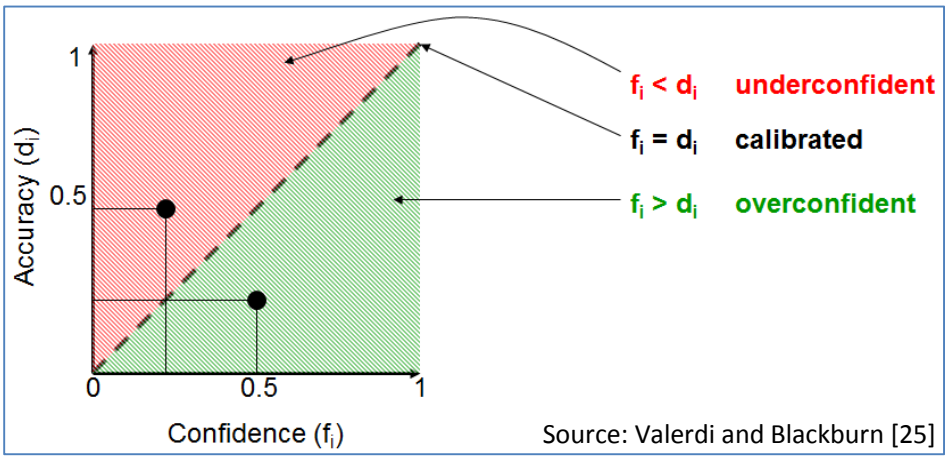


*Figure 25: Over and Under Confidence as Measured by a Brier Score*

We anticipate wide use of such binary test questions in training for participants in QUELCE-based estimation. True/false questions map very nicely to the QUELCE method where subject matter experts and estimators must make judgments (e.g., about appropriate change drivers, likely conditional probabilities, and future scenarios).[15]

We are continuing our research focus on methods to improve individual judgment skills such that the participants are able to make more realistic judgments commensurate with the state of their knowledge. We plan to follow the individual training studies with short tests of skill retention over time. If possible we will increase the number and diversity of participants in these studies to enable wider generalizability and additional experimental treatments (e.g., on the effects of initial training using generic interest questions prior to the domain-specific training, increasing the number of test batteries, and augmenting the existing didactic guidance).

The next stage of this research will also focus on methods of reconciling differences in judgment among members of expert teams [11, 14, 15, 18, 19, 22, 23]. That research will compare algorithmic and group decision methods with respect to accuracy, recognition of uncertainty, and time required to resolve differences among team members. If possible we will examine whether or not the team members have previously participated in calibration training. Improving ways to handle reconciliation of individual differences is crucial for methods like QUELCE, especially when dealing with group dynamics among collections of disparate stakeholders. Additional future research may compare the accuracy and precision of group decisions with that of individuals who are exceptionally skilled in making realistic judgments under uncertain conditions.

[15] Brier scores typically are used for the binary case (e.g., with true-false questions). However, Brier's original definition is applicable to the multinomial case. Hence it can handle multi-category measures such as those used in populating the QUELCE cause-effect matrix with a subset of the larger number of change drivers identified by subject matter experts.

# Appendix A: The Domain-Specific Test Batteries

Reduced size facsimiles of the three test batteries follow on the pages below.

## Domain-specific test battery 1

| # | Project<br>Answers must be entered as numbers only - .5 = 1/2 (only characters 0 to 9 and , $ . or - accepted) | Question | 90% Confidence Interval | |
|---|---|---|---|---|
| | | | Lower Bound | Upper Bound |
| 1 | **Apache JAMES Project:** A complete and portable enterprise mail engine based on open protocols; also a mail application platform that allows processing emails, e.g., to generate automatic replies, update databases, filter spam, or build message archives. | What is the project's current codebase size in LOC? | | |
| 2 | **LibreOffice:** A multi-platform, integrated office suite based on copyleft licenses and compatible with most document formats and standards: Includes spreadsheet, word processor, chart, business productivity, presentation, database, linix, C++ and other applications. | How much total effort in person years has been spent on this project? | | |
| 3 | **WebKit:** An open source web browser engine, the project`s HTML and JavaScript code began as a branch of the KDE (K Desktop Environment) libraries.WebKit is also the name of the engine used by Safari, Dashboard, Mail, and many other OS X applications.KDE is a GUI-based user interface primarily for Unix and Linux machines, but also available for Windows and Macintosh. | What is the current codebase size in LOC? | | |
| 4 | **TkCVS** is a Tcl/Tk-based graphical interface to the CVS and Subversion configuration management systems. It will also help with RCS. The user interface is consistent across Unix/Linux, Windows, and MacOS X. TkDiff is included for browsing and merging your changes. | How much total effort in person years has been spent on this project? | | |

| | | | | |
|---|---|---|---|---|
| 5 | **MySQL**, the most popular Open Source SQL database management system, is developed, distributed, and supported by Oracle Corporation. | What is the current code-base size in LOC? | | |
| 6 | **OpenGroupware.org** is a set of applications for contact, appointment, project, and content management. It is comparable to Exchange and SharePoint portal servers. It is accessible using Web interfaces and various native clients, including Outlook. Its servers run on almost any GNU/Linux system, can synchronize with Palm PDAs, and are completely scriptable using XML-RPC. | What is the current code-base size in LOC? | | |
| 7 | **Epiphany** is the web browser for the GNOME desktop. GNOME (GNU Network Object Model Environment) runs on Unix-like operating systems, most notably Linux. Powered by the WebKit engine, Epiphany aims to provide an uncomplicated user interface that enables users to focus on Web content instead of the browser application. | How much total effort in person years has been spent on this project? | | |
| 8 | **SVK** is a distributed version control system designed from the ground up to integrate cleanly with Subversion, the emerging standard in enterprise version control. With SVK, advanced branching and merging and even offline commits are easy. | How much total effort in person years has been spent on this project? | | |
| 9 | **Ingres** is an industrial strength database that is focused on reliability, security, scalability, and ease of use. It contains features demanded by the enterprise while providing the flexibility of open source. Its technology forms the foundation for numerous other industry-leading RDBMS systems. | What is the current code-base size in LOC? | | |

| | | | | |
|---|---|---|---|---|
| 10 | **WebCalendar** is a Web-based calendar application that can be configured as a single-user calendar, a multi-user calendar for groups of users, or as an event calendar viewable by visitors. WebCalendar requires a database such as MySQL, Oracle, PostgreSQL, MS SQL Server, ODBC, or Interbase. Features include email reminders, iCal/vCal import/export, remote subscriptions for Sunbird or Apple iCal, LDAP and NIS support, and translations for 29 languages. | How much total effort in person years has been spent on this project? | | |

**Domain-specific test battery 2** (Answer categories removed in the interest of space.)

| # | **Project**<br>Answers must be entered as numbers only - .5 = 1/2<br>(only characters 0 to 9 and , $ . or - accepted) | **Question** |
|---|---|---|
| 1 | **Mozilla Thunderbird:** Safe, fast, and easy email, with intelligent spam filters, quick message search, and customizable views.<br>---<br>Very large, active development team: 160 developers contributed new code over the past 12 months. Over the entire history of the project, 619 developers have contributed. The first lines of source code were added in 1998.<br>---<br>C++ = 46%; JavaScript = 21%; XML = 12%; Java = 6%; CSS = 6%; C = 5%; Other = 4%<br>31% comment to code ratio \| 323 person years of effort | What is the project's current codebase size in LOC? |
| 2 | **Calligra Suite:** A free, integrated work applications suite, build on top of KDE and Qt for use on Linux Desktop, Windows, Mac OS X and mobile phones: Includes a frame-based word processor, spreadsheet, presentation, flowchart & diagram, vector drawing, layered pixel image manipulation, & project management/ planning applications.<br>---<br>Very large, active development team: 83 developers contributed new code over the past twelve months. This is one of the largest open-source teams in the world. Over the entire history of the project, 491 developers have contributed. The first lines of source code were added in 1998.<br>---<br>C++ = 98%; C = 2%; Other < 1%<br>LOC = 1,173,122 \| 31% comment to code ratio | How much total effort in person years has been spent on this project? |
| 3 | **Google Chrome:** The open-source project behind Google Chrome (Chromium) builds on components from other open source software projects, including WebKit and Mozilla: It is aimed at improving stability, speed and security with a simple and efficient user interface.<br>---<br>Established codebase: The first lines of source code were added in 2008. The project has seen a substantial increase in activity over the last twelve months.<br>---<br>C++ = 39%; C = 33%; XML = 8%; HTML = 6%; Other =14%<br>LOC = 5,535,674 \| 1683 person years of effort | What is the ratio (%) of comments to LOC in the current codebase? |

| | | |
|---|---|---|
| 4 | **Mercurial** is a fast, lightweight Source Control Management system designed for efficient handling of very large distributed projects.<br>---<br>Over the past twelve months, 130 developers contributed new code. This is one of the largest open-source teams in the world, and is in the top 2% of all project teams in our database. Over the entire history of the project, 458 developers have contributed. The first lines of source code were added in 2005.<br>---<br>LOC = 152,551 \| 14% comment to code ratio \| 39 person years of effort | What percentage of the code is written in the product's major language (Perl)? |
| 5 | **PostgreSQL** is a powerful, open source relational database system. It has more than years of active development and a proven architecture that has earned it a strong reputation for reliability, data integrity, and correctness. It runs on all major operating systems, including Linux, UNIX (AIX, BSD, HP-UX, SGI IRIX, Mac OS X, Solaris, Tru64), and Windows.<br>---<br>Over the past twelve months, 13 developers contributed new code. The first lines of source code were added in . Well-commented source code, which could be a sign of a disciplined development team.<br>---<br>C = 87%; SQL = 7%; Other = 6%<br>LOC = 648,384 \| 37% comment to code ratio \| 179 person years of effort | In what year were the first lines of source code added? |
| 6 | **Buni Meldware Communication Suite:** Buni is a community of open source software developers and users dedicated to the research and development of communication and collaboration software. The Meldware Communications Suite includes Mail, a Calendar Server, Webmail, and a Secure Administration System.<br>---<br>During the past twelve months, this project had only one active contributor. Over the entire history of the project, 14 developers have contributed. The first lines of source code were added in 2003. Over the last twelve months, the project has seen a substantial decline in development activity. This could mean many things. Interest in this project may be waning, or it may indicate a maturing software base that requires fewer fixes.<br>---<br>Java = 69%; Actionscript = 15%; XML = 8%; Other = 8%<br>37 person years of effort | What is the project's current codebase size in LOC? |

| | | |
|---|---|---|
| 7 | **Camino:** A free, full featured, open source, GUI-based Web browser specifically designed for the Mac OS X operating system, Camino is based on Mozilla`s Gecko layout engine. It uses the OS X Aqua user interface and integrates a number of Mac OS X services and features, notably including password management, scanning available bookmarks, an integrated overview for managing multiple tabbed browsing, pop-up and ad blockers.<br>---<br>Mature, well-established codebase: The first lines of source code were added in 2002. The project has seen a substantial decline in activity over the last twelve months. This could mean many things. For example interest in the project may be waning, or a maturing software base may require fewer fixes.<br>---<br>C++ = 50%; C = 14%; XML = 13%; Objective-C = 9%; shell script = 5%; Other = 9%<br>LOC = 203,601 \| 23% comment to code ratio | How much total effort in person years has been spent on this project? |
| 8 | **Concurrent Versions System (CVS)** is a version control system, an important component of Source Configuration Management (SCM).<br>---<br>Over the past twelve months, only 2 developers contributed new code, making this a relatively small project. Over the entire history of the project, 31 developers have contributed. The first lines of source code were added in 1994. This is a relatively long time for an open source project to stay active, which might indicate a mature, relatively bug-free code base or a well-organized development team.<br>---<br>C = 47%; Autoconf = 22%; shell script =14%; Make = 5%; Other = 12%<br>LOC = 267,186 \| 68 person years of effort | What is the ratio (%) of comments to LOC in the current code-base? |
| 9 | **CUBRID** is a comprehensive open source relational database management system highly optimized for Web Applications. It includes JDBC, CSQL for command line administration, PHP & Ruby Libraries to connect to CUBRID.<br>---<br>Over the past twelve months, 13 developers contributed new code. Over the entire history of the project, 24 developers have contributed. CUBRID Database Management System has seen a substantial increase in activity over the last twelve months. This is probably a good sign that interest in this project is rising,<br>---<br>LOC = 1,189,422 \| 20% comment to code ratio \| 332 person years of effort | What percentage of the code is written in the product's major language (c)? |

| | | |
|---|---|---|
| 10 | **Mozilla Calendar project** develops Mozilla Sunbird (a stand-alone calendar application) and Lightning, a calendaring extension for Mozilla Thunderbird. Their goal is to bring Mozilla-style ease-of-use to your calendar, without tying you to a particular storage solution.<br>----------<br>Over the past twelve months, 157 developers contributed new code to Mozilla Calendar. This is one of the largest open-source teams in the world, and is in the top 2% of all project teams in our database. Over the entire history of the project, 495 developers have contributed. The first lines of source code were added in .<br>----------<br>C++ = 32%; JavaScript = 29%; XML = 15%; C = 7%; CSS = 7%; Java = 5%; Other = 5%<br>LOC = 927,266 32% comment to code ratio 253 person years of effort | In what year were the first lines of source code added? |

**Domain-specific test battery 3** (Answer categories removed in the interest of space.)

| # | **Project**<br>Answers must be entered as numbers only - .5 = 1/2<br>(only characters 0 to 9 and , $ . or - accepted) | **Question** |
|---|---|---|
| 1 | **Buni Meldware Communication Suite:** Mail, a Calendar Server, Webmail, and a Secure Administration System<br>---<br>A substantial decline in development activity over the last twelve months: Over the entire history of the project, 14 developers have contributed. The first lines of source code were added in 2003.<br>---<br>Java = 69%; ActionScript = 15%; XML = 8%; Other = 8%<br>37 person years of effort | What is the ratio (%) of comments to LOC in the current codebase? |
| 2 | **NeoOffice:** A fully-featured set of office applications based on the OpenOffice.org office suite that includes word processing, spreadsheet, presentation, and drawing programs for Mac OS X that can import, edit, and exchange files with other popular office programs<br>---<br>Mature, well-established codebase: The first lines of source code were added in 2003. During the past twleve months, this project had only one active contributor. Very few source code comments: only14% of the C++ code.<br>---<br>C++ = 84%; XML = 6%; Objective-C = 5%; Other = 5%<br>LOC = 392,932 \| 17% comment to code ratio | How much total effort in person years has been spent on this project? |
| 3 | **Apache HTTP Server:** A feature-rich Web server with freely-available source code: Includes FTP, caching, Common Gateway Interface (CGI), dynamic content, authentication, intranet, plugin, xml, SSL, authorization, modular and proxy functionality.<br>---<br>Very large, active development team: Over the past twelve months, 33 developers contributed new code. This is one of the largest open-source teams in the world. The first lines of source code were added in 1996. Its well-commented source code could be a sign of a disciplined development team.<br>---<br>XML = 64%; C = 28%; forth = 5%; Other = 3%<br>LOC = 1,547,962 \| 440 person years of effort | What is the ratio (%) of comments to LOC in the current codebase? |
| 4 | **Apache Continuum** is a continuous integration server for building Java based projects. It supports a wide range of projects.<br>---<br>There has been a substantial decline in development activity over the last twelve months. This could mean many things. Interest in this project may be waning, or it may indicate a maturing software base that requires fewer fixes. The first lines of source code were added in 2005.<br>---<br>LOC = 484,842 \| 24% comment to code ratio \| 128 person years of effort | What percentage of the code is written in the product's major language (Java)? |

| | | |
|---|---|---|
| 5 | **Firebird** is a relational database offering many ANSI SQL standard features that runs on Linux, Windows, and a variety of Unix platforms. It offers excellent concurrency, high performance, and powerful language support for stored procedures and triggers. It has been used in production systems, under a variety of names .<br>----------<br>Over the past twelve months, 16 developers contributed new code. This is a relatively large team, putting this project among the top 10% of all project teams in our database. Over the entire history of the project, 68 developers have contributed. The first lines of source code were added in .<br>There has been a substantial decline in development activity over the last twelve months; however this could mean many things. Interest in this project may be waning, or a maturing software base may require fewer fixes.<br>----------<br>C = 44%; C++ = 24%; XML = 11%; Other = 21%<br>LOC = 4,028,411 \| 19% comment to code ratio \| 1190 person years of effort | In what year were the first lines of source code added? |
| 6 | **OBM** is a groupware, email, LDAP, Windows PDC, CRM, and project management application. It is mainly used as an Exchange or Notes/Domino groupware and mail server replacement, as an LDAP directory, as a Windows PDC, as a contact and customer database, as a project management tool, or as any combination of these functions. It provides groupware (calendars, contacts, and tasks) connectors for Outlook, Thunderbird/Lightning, and PDAs. It supports internationalization and themes. Highly scalable. It is used by sites from five to many thousands of users.<br>----------<br>Over the past twelve months, 32 developers contributed new code. This is one of the largest open-source teams in the world, and is in the top 2% of all projects in our database. The first lines of source code were added in 2002.<br>----------<br>PHP = 53%; Java = 18%; SQL = 10%; Perl = 5%; Javascript = 5%; Other = 9%<br>LOC = 849,261 \| 26% comment to code ratio | How much total effort in person years has been spent on this project? |
| 7 | **Mozilla Firefox:** A full featured Web browser: With more than 15,000 improvements, version 3 is faster, more secure, and fully customizable.<br>----------<br>Very large, active development team: Over the past twelve months, 706 developers contributed new code. This is one of the largest open-source teams in the world. The first lines of source code were added in 2002.<br>----------<br>C++ = 38%; C = 19%; JavaScript = 14%; HTML = 9%; XML = 7%; Other = 13%<br>28% comment to code ratio \| 2002 person years of effort | What is the project's current codebase size in LOC? |

| | | |
|---|---|---|
| 8 | **Bugzilla** is a web-based bug tracking tool. It works with an existing web server, e.g. Apache, and with an existing SQL database, e.g. MySQL or PostgreSQL.<br>---<br>Over the past twelve months, 33 developers contributed new code. This is one of the largest open-source teams in the world, and is in the top 2% of all project teams our database. Over the entire history of the project, 102 developers have contributed. The first lines of source code were added in 1998.<br>---<br>Perl = 77%; XML = 15%; Other = 8%<br>LOC = 69,900 \| 17 person years of effort | What is the ratio (%) of comments to LOC in the current code-base? |
| 9 | **BlackRay** is a relational database system designed to offer performance features commonly associated with search engines. It offers SQL support and sophisticated operational and management features. Load-balancing and operational stability by means of N+1 redundance are included. It is a hybrid, offering transaction support, data-versioned snapshots, and sophisticated function-based indices. Wildcards, phonetic, and fuzzy logic searches are supported, as well.<br>---<br>This is a small development team. Over the past twelve months, only 2 developers contributed new code. Over the entire history of the project, 7 developers have contributed. There has been a substantial decline in development activity over the past twelve month; however this could mean many things. Interest in this project may be waning, or it a maturing software base may require fewer fixes. Well-commented source code puts this project among the highest one-third of all C++ projects in our database.<br>---<br>LOC = 119,867 \| 42% comment to code ratio \| 30 person years of effort | What percentage of the code is written in the product's major language (C++)? |
| 10 | The **Calendar and Contacts Server** project is a standards-compliant server implementing the CalDAV and CardDAV protocols. It provides a shared location on the network allowing multiple users to store and edit calendaring and contact information.<br>---<br>The first lines of source code were added in . This is a relatively long time for an open source project to stay active, and can be a very good sign. It might indicate a mature and relatively bug-free code base, and can be a sign of an organized, dedicated development team. This high number of comments puts Calendar and Contacts Server among the highest one-third of all Python projects in our database.<br>---<br>Python = 82%; XML = 5%; Other = 3%<br>LOC = 144,741 \| 51% comment to code ratio \| 36 person years of effort | In what year were the first lines of source code added? |

# Appendix B: The Domain-Specific Reference Points for Test Battery 3

The same reference points were used for both test battery 2 and test battery 3. Notice that the correct answers to the test battery 3 questions remained hidden from the participants' view. The correct answers to both test batteries 2 and 3 were hidden from view while the participants answered the questions in test battery 2.

A reduced size facsimile of the hard copy reference points follows.

| Mailers | | | | | |
|---|---|---|---|---|---|
| Summary | Languages | Ratio of Comments to Code | Codebase Size in LOC | Description of Development Team | Total Effort |
| **Mozilla Thunderbird:** Safe, fast, and easy email, with intelligent spam filters, quick message search, and customizable views | C++ = 46%<br>JavaScript = 21%<br>XML = 12%<br>Java = 6%<br>CSS = 6%<br>C = 5%<br>Other = 4% | 31% | 1,132,278 | Very large, active development team: 160 developers contributed new code over the past 12 months. Over the entire history of the project, 619 developers have contributed. The first lines of source code were added in 1998. | 323 person years |
| **Bongo:** An easy-to-use mail and calendar system, offering a simple yet powerful user interface | PHP = 26%<br>C = 22%<br>JavaScript = 20%<br>HTML = 16%<br>XML = 7%<br>Other = 9% | 17% | 478,824 | A substantial decline in development activity over the last twelve months: This could mean many things. For example interest in the project may be waning, or a maturing software base may require fewer fixes. The first lines of source code were added in 2007. | 127 person years |
| **Apache JAMES Project**: A complete and portable enterprise mail engine based on open protocols; also a mail application platform that allows processing emails, e.g., to generate automatic replies, update databases, filter spam, or build message archives | Java = 55%<br>HTML = 26%<br>XML = 14%<br>Other = 5% | 46% | 3,388,490 | Large, active development team: 10 developers contributed new code over the past 12 months, which is a substantial increase in recent activity. The first lines of source code were added in 1998, and 619 developers have contributed over the entire history of the project. Well-commented source code, which could be a sign of a disciplined development team. | ??? |
| **Buni Meldware Communication Suite**: Mail, a Calendar Server, Webmail, and a Secure Administration System | Java = 69%<br>ActionScript = 15%<br>XML = 8%<br>Other = 8% | | 146,695 | A substantial decline in development activity over the last twelve months: Over the entire history of the project, 14 developers have contributed. The first lines of source code were added in 2003. | 37 person years |

| Office Suites | | | | | |
|---|---|---|---|---|---|
| Summary | Languages | Ratio of Comments to Code | Codebase Size in LOC | Description of Development Team | Total Effort |
| **Apache OpenOffice:** A multiplatform and multilingual office suite and an open-source project that is compatible with all major office suites | C++ = 74%<br>XML = 12%<br>Java = 5%<br>Other = 9% | 25% | 15,110,141 | Very large, active development team: Over the past twelve months, 23 developers contributed new code. Over the entire history of the project, 332 developers have contributed. The first lines of source code were added in 2000. A substantial decline in development activity over the last twelve months: This could mean many things. For example interest in the project may be waning, or a maturing software base may require fewer fixes. | 4777 person years |
| **LibreOffice:** A multi-platform, integrated office suite based on copyleft licenses and compatible with most document formats and standards: Includes spreadsheet, word processor, chart, business productivity, presentation, database, linix, C++ and other applications. | C++ = 78%<br>XML = 8%<br>Java = 5%<br>Other = 9 | 27% | 12,767,297 | Very large, active development team: 316 developers contributed new code over the past 12 months, which is a substantial decline in development activity. This is one of the largest open-source teams in the world. Over the entire history of the project, 844 developers have contributed. The first lines of source code were added in 2000. | 4021 person years |
| **Calligra Suite:** A free, integrated work applications suite, build on top of KDE and Qt for use on Linux Desktop, Windows, Mac OS X and mobile phones: Includes a frame-based word processor, spreadsheet, presentation, flowchart & diagram, vector drawing, layered pixel image manipulation, & project management/ planning applications. | C++ = 98%<br>C = 2%<br>Other < 1% | 31% | 1,173,122 | Very large, active development team: 83 developers contributed new code over the past twelve months. This is one of the largest open-source teams in the world. Over the entire history of the project, 491 developers have contributed. The first lines of source code were added in 1998. | 356 person years |
| **NeoOffice:** A fully-featured set of office applications based on the OpenOffice.org office suite that includes word processing, spreadsheet, presentation, and drawing programs for Mac OS X that can import, edit, and exchange files with other popular office programs | C++ = 84%<br>XML = 6%<br>Objective-C = 5%<br>Other = 5% | 17% | 392,932 | Mature, well-established codebase: The first lines of source code were added in 2003. During the past twleve months, this project had only one active contributor. Very few source code comments: only14% of the C++ code. | |
| **OxygenOffice Professional**: Functionality includes SQL, GUI, python, C++, unix, drawing, spreadsheet, database, painting, meeting, word-processor, presentation, chart, project, graphics media technology & a library of over 3,400 graphics, several templates, sample documents, and over 90 fonts. | C++ = 36%<br>Java = 33%<br>XML = 24%<br>Other = 7% | 13% | 61,084 | Established codebase: The first lines of source code were added 2006. Over the past twelve months, only 2 developers contributed new code, which is a substantial decline in development activity. Very few source code comments: only 6% of the C++ code. | 15 person years |

| Browsers | | | | | |
|---|---|---|---|---|---|
| Summary | Languages | Ratio of Comments to Code | Codebase Size in LOC | Description of Development Team | Total Effort |
| **Mozilla Firefox**: A full featured Web browser: With more than 15,000 improvements, version 3 is faster, more secure, and fully customizable. | C++ = 38%; C = 19%; JavaScript = 14%; HTML = 9%; XML = 7%; Other = 13% | 28% | | Very large, active development team: Over the past twelve months, 720 developers contributed new code. This is one of the largest open-source teams in the world. The first lines of source code were added in 2002. | 2002 person years |
| **Google Chrome**: The open-source project behind Google Chrome (Chromium) builds on components from other open source software projects, including WebKit and Mozilla: It is aimed at improving stability, speed and security with a simple and efficient user interface. | C++ = 39%; C = 33%; XML = 8%; HTML = 6%; Other =14% | 20% | 5,535,674 | Established codebase: The first lines of source code were added in 2008. The project has seen a substantial increase in activity over the last twelve months. | 1683 person years |
| **WebKit**: An open source web browser engine, the project's HTML and JavaScript code began as a branch of the KDE (K Desktop Environment) libraries.WebKit is also the name of the engine used by Safari, Dashboard, Mail, and many other OS X applications.KDE is a GUI-based user interface primarily for Unix and Linux machines, but also available for Windows and Macintosh. | HTML = 34%; C++ = 31%; JavaScript = 17%; XML = 6%; Other = 12% | 20% | 4,103,073 | Very large, active development team: 270 developers contributed new code over the past twelve months. This is one of the largest open-source teams in the world. The first lines of source code were added in 2001. | 1216 person years |
| **Epiphany** is the web browser for the GNOME desktop. GNOME (GNU Network Object Model Environment) runs on Unix-like operating systems, most notably Linux. Powered by the WebKit engine, Epiphany aims to provide an uncomplicated user interface that enables users to focus on Web content instead of the browser application. | C = 91%; XML = 6%; Other = 3% | 16% | 57,630 | Very large, active development team: 115 developers contributed new code over the past twelve months, which is a substantial increase in activity. This is one of the largest open-source teams in the world. The first lines of source code were added in 2002. | 13 person years |
| **Camino**: A free, full featured, open source, GUI-based Web browser specifically designed for the Mac OS X operating system, Camino is based on Mozilla's Gecko layout engine. It uses the OS X Aqua user interface and integrates a number of Mac OS X services and features, notably including password management, scanning available bookmarks, an integrated overview for managing multiple tabbed browsing, pop-up and ad blockers. | C++ = 50%; C = 14% XML = 13%; Objective-C = 9% shell script = 5% Other = 9% | 23% | 203,601 | Mature, well-established codebase: The first lines of source code were added in 2002. The project has seen a substantial decline in activity over the last twelve months. This could mean many things. For example interest in the project may be waning, or a maturing software base may require fewer fixes. | 51 person years |
| **XULRunner** is a Mozilla runtime package that can be used to bootstrap XUL+XPCOM applications that are as rich as Firefox and Thunderbird. It will provide mechanisms for installing, upgrading, and uninstalling these applications. XULRunner will also provide libxul, a solution which allows the embedding of Mozilla technologies in other projects and products. | C++ = 58%; Make = 23%; shell script = 9%; XML=7%; Other=3% | 61% | 2,399 | The source code for XULRunner has not been changed in over a year. Over the entire history of the project, 22 developers have contributed. The first lines of source code were added in 2004. This is a relatively long time for an open source project to stay active, which might indicate a mature, relatively bug-free code base or a well-organized development team. | ??? |
| **Konqueror** is a file manager, web browser and file viewer, which was developed as part of the K Desktop Environment (KDE) by volunteers and runs on most Unix-like operating systems. | C++ = 92% other= 8% | 22% | 34,144 | The first lines of source code were added to Konqueror in 2007. This is a relatively long time for an open source project to stay active. Over the last twelve months, Konqueror has seen a substantial decline in development activity. | 8 person years |
| **ExpressoBrowser** is designed to be a simple fast web browser using WebKit Cairo and JavaScriptCore rendering engine with a C# wrapper. The project aims to eventually support Windows, Mac and Linux using native user interface calls (Win.Forms/Aero, Cocoa# and GTK#). | C++ = 80% C# = 12% C = 5% Other = 3% | 16% | 34,078 | Over the past twelve months, only 2 developers contributed new code to ExpressoBrowser, making this a relatively small project. The source code for ExpressoBrowser has less than a year of continuous activity, making this a relatively new project. A short history is not necessarily a bad thing, but it can indicate an immature or volatile project. However the source code might actually be older. Many projects begin by duplicating a large amount of source code from an existing, older project. Only 6% of the source code lines are comments, which is among the lowest 10% of all C++ projects on our database. | 8 person years |
| **Apache HTTP Server:** A feature-rich Web server with freely-available source code: Includes FTP, caching, Common Gateway Interface (CGI), dynamic content, authentication, intranet, plugin, xml, SSL, authorization, modular and proxy functionality. | XML = 64% C = 28% forth = 5% Other = 3% | | 1,547,962 | Very large, active development team: Over the past twelve months, 33 developers contributed new code. This is one of the largest open-source teams in the world. The first lines of source code were added in 1996. Its well-commented source code could be a sign of a disciplined development team. | 440 person years |

| Configuration Management | | | | | |
|---|---|---|---|---|---|
| Summary | Languages | Ratio of Comments to Code | Codebase Size in LOC | Description of Development Team | Total Effort |
| **Concurrent Versions System (CVS)** is a version control system, an important component of Source Configuration Management (SCM). | C = 47%<br>Autoconf = 22%<br>shell script =14%<br>Make = 5%<br>Other = 12% | 23% | 267,186 | Over the past twelve months, only 2 developers contributed new code, making this a relatively small project. Over the entire history of the project, 31 developers have contributed. The first lines of source code were added in 1994. This is a relatively long time for an open source project to stay active, which might indicate a mature, relatively bug-free code base or a well-organized development team. | 68 person years |
| TkCVS is a Tcl/Tk-based graphical interface to the CVS and Subversion configuration management systems. It will also help with RCS. The user interface is consistent across Unix/Linux, Windows, and MacOS X. TkDiff is included for browsing and merging your changes. | Tcl = 87%<br>shell script =12%<br>Other = 1% | 17% | 25,225 | During the past twelve months, this project had only one active contributor. About one-third of all active projects in our database are solo efforts. Over the entire history of the project, 4 developers have contributed. The first lines of source code were in 1994. This is a relatively long time for an open source project to stay active. | 6 person years |
| **Apache Continuum** is a continuous integration server for building Java based projects. It supports a wide range of projects. | | 24% | 484,842 | There has been a substantial decline in development activity over the last twelve months. This could mean many things. Interest in this project may be waning, or it may indicate a maturing software base that requires fewer fixes. The first lines of source code were added in 2005. | 128 person years |
| **SVK** is a distributed version control system designed from the ground up to integrate cleanly with Subversion, the emerging standard in enterprise version control. With SVK, advanced branching and merging and even offline commits are easy. | Perl= 95%<br>Other = 5% | 33% | 38,544 | During the past twelve months, this project had only one active contributor. Over the entire history of the project, 25 developers have contributed. The first lines of source code were added in 2004. | 9 person years |
| **Mercurial** is a fast, lightweight Source Control Management system designed for efficient handling of very large distributed projects. | Perl = 53%<br>Python = 41%<br>Other = 6% | 14% | 152,551 | Over the past twelve months, 130 developers contributed new code. This is one of the largest open-source teams in the world, and is in the top 2% of all project teams in our database. Over the entire history of the project, 458 developers have contributed. The first lines of source code were added in 2005. | 39 person years |
| Bugzilla is a web-based bug tracking tool. It works with an existing web server, e.g. Apache, and with an existing SQL database, e.g. MySQL or PostgreSQL. | Perl = 77%<br>XML = 15%<br>Other = 8% | | 69,900 | Over the past twelve months, 33 developers contributed new code. This is one of the largest open-source teams in the world, and is in the top 2% of all project teams our database. Over the entire history of the project, 102 developers have contributed. The first lines of source code were added in 1998. | 17 person years |
| rdesktop is an open source client for the Remote Desktop Protocol (RDP), which is used in a number of Microsoft products including Windows NT Terminal Server, Windows 2000 Server, Windows XP, and Windows 2003 Server. rdesktop currently runs on Linux and other Unix based platforms with the X Window System. | C = 74%<br>C++ = 14<br>shell script = 5%<br>Other = 7% | 12% | 114,404 | There has been a substantial decline in development activity over the last twelve months. The first lines of source code were added to rdesktop in 2000. | 29 person years |

| Databases | | | | | |
|---|---|---|---|---|---|
| Summary | Languages | Ratio of Comments to Code | Codebase Size in LOC | Description of Development Team | Total Effort |
| **MySQL**, the most popular Open Source SQL database management system, is developed, distributed, and supported by Oracle Corporation. | C++ = 59%<br>C = 35%<br>Other = 6% | 23% | 1,394,704 | 117 developers contributed new code over the past twelve months. This is one of the largest open-source teams in the world, and is in the top 2% of all project teams our database. Over the entire history of the project, 1182 developers have contributed. The first lines of source code were added in 2000. This is a relatively long time for an open source project to stay active. | 400 person years |
| **HyperSQL** Database is a relational database engine written in Java with a JDBC driver that supports a subset of ANSI SQL: 1999. It offers a small, fast database engine. Embedded and server modes are available. It includes tools such as a minimal Web server, in-memory query and management tools (which can be run as applets or servlets, too), a test framework, PHP compatibility, Eclipse and NetBeans IDE compatibility, and a number of demonstration examples. | Java = 51%<br>JavaScript = 11%<br>XML = 9%<br>HTML = 9%<br>C# = 7%<br>C = 7%<br>Other = 6% | 37% | 567,023 | Three developers contributed new code over the past twelve months; 11 developers have contributed code over the entire history of the project. The first lines of source code were added 2002. This is a relatively long time for an open source project to stay active, which might indicate a mature, relatively bug-free code base or a well-organized development team. | 152 person years |
| **PostgreSQL** is a powerful, open source relational database system. It has more than 15 years of active development and a proven architecture that has earned it a strong reputation for reliability, data integrity, and correctness. It runs on all major operating systems, including Linux, UNIX (AIX, BSD, HP-UX, SGI IRIX, Mac OS X, Solaris, Tru64), and Windows. | C = 87%<br>SQL = 7%<br>Other = 6% | 37% | 648,384 | Over the past twelve months, 13 developers contributed new code. The first lines of source code were added in 1996. Well-commented source code, which could be a sign of a disciplined development team. | 179 person years |
| **Firebird** is a relational database offering many ANSI SQL standard features that runs on Linux, Windows, and a variety of Unix platforms. It offers excellent concurrency, high performance, and powerful language support for stored procedures and triggers. It has been used in production systems, under a variety of names . It is based on the source code released by Inprise Corp (now known as Borland Software Corp) . | C = 44%<br>C++ = 24%<br>XML = 11%<br>Other = 21% | 19% | 4,028,411 | Over the past twelve months, 16 developers contributed new code. This is a relatively large team, putting this project among the top 10% of all project teams in our database. Over the entire history of the project, 68 developers have contributed. The first lines of source code were added . There has been a substantial decline in development activity over the last twelve months; however this could mean many things. Interest in this project may be waning, or a maturing software base may require fewer fixes. | 1190 person years |
| **Ingres** is an industrial strength database that is focused on reliability, security, scalability, and ease of use. It contains features demanded by the enterprise while providing the flexibility of open source. Its technology forms the foundation for numerous other industry-leading RDBMS systems. | C = 55%<br>XML = 16%<br>C++ = 12%<br>Scheme = 6%<br>Other = 11% | 50% | 3,750,002 | Over the past twelve months, 34 developers contributed new code. This is one of the largest open-source teams in the world, and is in the top 2% of all project teams in our database. Over the entire history of the project, 71 developers have contributed. The first lines of source code were added in 2008.Extremely well-commented source code puts this project among the top 10% of all C projects in our database. There has been a substantial decline in development activity over the last twelve months. | 1106 person years |
| **CUBRID** is a comprehensive open source relational database management system highly optimized for Web Applications. It includes JDBC, CSQL for command line administration, PHP & Ruby Libraries to connect to CUBRID. | C = 69%<br>Autoconf = 8%<br>C++ = 8%<br>shell script = 7%<br>Other = 8% | 20% | 1,189,422 | Over the past twelve months, 13 developers contributed new code. Over the entire history of the project, 24 developers have contributed. CUBRID Database Management System has seen a substantial increase in activity over the last twelve months. This is probably a good sign that interest in this project is rising, | 332 person years |
| **BlackRay** is a relational database system designed to offer performance features commonly associated with search engines. It offers SQL support and sophisticated operational and management features. Load-balancing and operational stability by means of N+1 redundance are included. It is a hybrid, offering transaction support, data-versioned snapshots, and sophisticated function-based indices. Wildcards, phonetic, and fuzzy logic searches are supported, as well. | | 42% | 119,867 | This is a small development team. Over the past twelve months, only 2 developers contributed new code. Over the entire history of the project, 7 developers have contributed. There has been a substantial decline in development activity over the past twelve month; however this could mean many things. Interest in this project may be waning, or it a maturing software base may require fewer fixes. Well-commented source code puts this project among the highest one-third of all C++ projects in our database. | 30 person years |

| Calendars & Groupware | | | | | |
|---|---|---|---|---|---|
| Summary | Languages | Ratio of Comments to Code | Codebase Size in LOC | Description of Development Team | Total Effort |
| **WebCalendar** is a Web-based calendar application that can be configured as a single-user calendar, a multi-user calendar for groups of users, or as an event calendar viewable by visitors. WebCalendar requires a database such as MySQL, Oracle, PostgreSQL, MS SQL Server, ODBC, or Interbase. Features include email reminders, iCal/vCal import/export, remote subscriptions for Sunbird or Apple iCal, LDAP and NIS support, and translations for 29 languages. | PHP = 50%; JavaScript = 19%; HTML = 15%; Java = 5%; SQL = 5%; Other = 6% | 22% | 68,915 | Over the past twelve months, only 3 developers contributed new code. Over the entire history of the project, 7 developers have contributed. The first lines of source code were added in 2001. This is a relatively long time for an open source project to stay active, which might indicate a mature and relatively bug-free code base, and can be a sign of an organized, dedicated development team. | 17 person years |
| **Mozilla Calendar project** develops Mozilla Sunbird (a stand-alone calendar application) and Lightning, a calendaring extension for Mozilla Thunderbird. Their goal is to bring Mozilla-style ease-of-use to your calendar, without tying you to a particular storage solution. | C++ = 32%; JavaScript = 29%; XML = 15%; C = 7%; CSS = 7%; Java = 5%; Other = 5% | 32% | 927,266 | Over the past twelve months, 157 developers contributed new code to Mozilla Calendar. This is one of the largest open-source teams in the world, and is in the top 2% of all project teams in our database. Over the entire history of the project, 495 developers have contributed. The first lines of source code were added in 1998. | 253 person years |
| **OBM** is a groupware, email, LDAP, Windows PDC, CRM, and project management application. It is mainly used as an Exchange or Notes/Domino groupware and mail server replacement, as an LDAP directory, as a Windows PDC, as a contact and customer database, as a project management tool, or as any combination of these functions. It provides groupware (calendars, contacts, and tasks) connectors for Outlook, Thunderbird/Lightning, and PDAs. It supports internationalization and themes. Highly scalable, it is used by sites from five to many thousands of users. | PHP = 53%; Java = 18%; SQL = 10%; Perl = 5%; JavaScript = 5%; Other = 9% | 26% | 849,261 | Over the past twelve months, 32 developers contributed new code. This is one of the largest open-source teams in the world, and is in the top 2% of all projects in our database. The first lines of source code were added in 2002. | |
| **OpenGroupware.org** is a set of applications for contact, appointment, project, and content management. It is comparable to Exchange and SharePoint portal servers. It is accessible using Web interfaces and various native clients, including Outlook. Its servers run on almost any GNU/Linux system, can synchronize with Palm PDAs, and are completely scriptable using XML-RPC. | Objective-C = 77%; Other = 23% | 19% | 757,141 | Over the last twelve months, OpenGroupware.org has seen a substantial increase in activity. This is probably good sign that interest in this project is rising The first lines of source code were added in 2004. This is a relatively long time for an open source project to stay active. During the past twelve months, this project had only one active contributor. About one-third of all active projects on our database are solo efforts. Over the entire history of the project, 12 developers have contributed. | 207 person years |
| **Buni Meldware Communication Suite**: Buni is a community of open source software developers and users dedicated to the research and development of communication and collaboration software. The Meldware Communications Suite includes Mail, a Calendar Server, Webmail, and a Secure Administration System. | Java = 69%; ActionScript = 15%; XML = 8%; Other = 8% | | 146,695 | During the past twelve months, this project had only one active contributor. Over the entire history of the project, 14 developers have contributed. The first lines of source code were added in 2003. Over the last twelve months, the project has seen a substantial decline in development activity. This could mean many things. Interest in this project may be waning, or it may indicate a maturing software base that requires fewer fixes. | 37 person years |
| The **Calendar and Contacts Server** project is a standards-compliant server implementing the CalDAV and CardDAV protocols. It provides a shared location on the network allowing multiple users to store and edit calendaring and contact information. | Python = 82%; XML = 5%; Other = 3% | 51% | 144,741 | The first lines of source code were added in . This is a relatively long time for an open source project to stay active, and can be a very good sign. It might indicate a mature and relatively bug-free code base, and can be a sign of an organized, dedicated development team. This high number of comments puts Calendar and Contacts Server among the highest one-third of all Python projects in our database. | 36 person years |

# Appendix C: Participant Feedback Questionnaire

A reduced size facsimile of the hard copy participant feedback questionnaire follows below and on the next page.

**Improving Expert Judgment for Software Engineering: About Today's Tasks**

1. How familiar are you with the kinds of software systems about which we asked today? *(Please select one)*
   - ☐ Very familiar
   - ☐ Familiar
   - ☐ Mixed familiarity across the questions
   - ☐ Unfamiliar
   - ☐ Very unfamiliar

2. Were you surprised about how well or poorly you did? *(Please check one)*
   - ☐ Very surprised
   - ☐ Somewhat surprised
   - ☐ It varied across the questions
   - ☐ Not surprised

3. How much difficulty did you have in answering the questions?
   *(Please check one – and describe briefly in the white space here)*
   - ☐ Very easy
   - ☐ Reasonably easy
   - ☐ It varied across the questions
   - ☐ Rather hard
   - ☐ Very Hard

4. How much guidance would you like to have had today? *(Please check one)*
   - ☐ A lot more
   - ☐ A little more
   - ☐ No more
   - ☐ A little less
   - ☐ A lot less

5. How much practice would you like to have had today? *(Please check one)*
   - ☐ A lot more
   - ☐ A little more
   - ☐ No more
   - ☐ A little less
   - ☐ A lot less

6. Which of the following methods did you use to match your intervals with the state of your knowledge?
   *(Please check as many as apply)*
   - ☐ Making "equivalent bets"
   - ☐ Thinking of pros and cons
   - ☐ Thinking of other related factors that might change your answers
   - ☐ Widening your intervals with limited time to think it through
   - ☐ Other ways *(Please describe briefly)*

7. How informative was the contextual information in the project descriptions shown with the questions? *(Please check one)*
   - ☐ Very informative
   - ☐ Reasonably informative
   - ☐ It varied across the questions
   - ☐ Rather uninformative
   - ☐ Very uninformative

8. How informative were the tables of "reference points" describing other projects along with the ones asked about in the questions? *(Please check one)*
   - ☐ Very informative
   - ☐ Reasonably informative
   - ☐ It varied across the questions
   - ☐ Rather uninformative
   - ☐ Very uninformative

9. How helpful were the contextual information and reference points?
   (Please check as many as apply)
   - ☐ Very helpful: I couldn't have done without them
   - ☐ More helpful than not
   - ☐ It varied across the questions
   - ☐ Less helpful than I would have liked
   - ☐ I often found them to be misleading

10. What other kinds of information did you use to inform your decisions? *(Please describe briefly)*

11. What else would you have liked to know? *(Please describe briefly)*

12. If you attended the session on Monday: How much do you think it helped you think through your answers to the questions today? *(Please check one)*
    - ☐ Very helpful
    - ☐ More helpful than not
    - ☐ It varied across the questions
    - ☐ Less helpful than I would have liked
    - ☐ Little if any help

**Thanks very much for your time and effort!**

# References/Bibliography

*URLs are valid as of the publication date of this document.*


1. Black, Thomas R. "Mann-Whitney U Test" in Michael S. Lewis-Beck, Alan Bryman, and Tim Futing Liao, *The SAGE Encyclopedia of Social Science Research Methods*. Sage Publications, 2004.

2. Brier, G. W. (1950). "Verification of Forecasts Expressed in Terms of Probability." *Monthly Weather Review*, *75*, 1-3.

3. Buehler, R., Griffin, D. and Ross, M. "Exploring the 'Planning Fallacy': Why People Underestimate their Task Completion Times," *Journal of Personality and Social Psychology, 67,* 1994.

4. Evans, D. *Risk Intelligence: How to Live with Uncertainty*, Free Press, 2012.

5. Fast, N., et al. "Power and Overconfident Decision-Making," *Organizational Behavior and Human Decision Processes*, 2011.

6. Ferguson, R., Goldenson, D., McCurley, J., Stoddard, R., Zubrow, D., & Anderson, D. (2011). *Quantifying Uncertainty in Early Lifecycle Cost Estimation (QUELCE)* (CMU/SEI-2011-TR-026). http://www.sei.cmu.edu/library/abstracts/reports/11tr026.cfm

7. Francesca, G. and Don, A. M. "Effects of Task Difficulty on Use of Advice." *Journal of Behavioral Decision Making 20*, 1: 21, 2007.

8. Freeman, L. C. *Elementary Applied Statistics for Students in Behavioral Science*. New York: John Wiley & Sons, 1965.

9. Gino, F., Sharek Z., et al. "Keeping the Illusion of Control Under Control: Ceilings, Floors, and Imperfect Calibration." *Organizational Behavior and Human Decision Processes 114*, 2: 104, 2011.

10. Healy, P, Moore, D. "Bayesian Overconfidence," *Social Science Research Network* working paper, 2007.

11. Hora, S. "Probability Judgments for Continuous Quantities: Linear Combinations and Calibration," *Management Science* 50, 5 (May 2004): 597-604.

12. Hubbard, D. and Evans, D. "Problems with Scoring Methods and Ordinal Scales in Risk Assessment," *IBM Journal of Research & Development, 54(3)*, May-June 2010.

13. Hubbard, D. *How to Measure Anything: Finding the Value of Intangibles in Business*. Hoboken, New Jersey: John Wiley & Sons, 2nd edition, 2010.

14. Jørgensen, M, et al. “Better Sure than Safe? Overconfidence in Judgment-Based Software Development Effort Prediction Intervals,” *Journal of Systems and Software,70,* 1-2 (February 2004):79-93.

15. Kitchens, L. J. *Basic Statistics and Data Analysis*. Cengage Learning, 2002.

16. Miranda, E. “Improving Subjective Estimations Using Paired Comparisons” *IEEE Software 18*, 1 (January 2001).

17. Moore, D., Cain, D. “Overconfidence and Underconfidence: When and Why People Underestimate (and Overestimate) the Competition,” *Organizational Behavior and Human Decision Processes*, 2007.

18. Moløkken-Østvold, K. and Jørgensen, M. “Group Processes in Software Effort Estimation.” *Empirical Software Engineering 9*: 315–334, 2004.

19. SEI Cost Estimation Research Group: R. Ferguson, D. Goldenson, J. McCurley, R. Stoddard, D. Zubrow. “An Innovative Approach to Quantifying Uncertainty in Early Lifecycle Cost Estimation.” *DACS Journal of Software Technology*, March 2012.

20. Shepperd, M. and Cartwright, M. “Predicting with Sparse Data.” *IEEE Transactions on Software Engineering,* 2001.

21. Tukey, John W. *Exploratory Data Analysis*. Addison-Wesley, 1977.

22. Ullman, David G. *Making Robust Decisions: Decision Management for Technical, Business, and Service Teams.* Victoria, BC Canada: Trafford Publishing, 2006.

23. Valerdi, R. “Optimism in Cost Estimation.” in *The IFPUG Guide to IT and Software Measurement*, edited by IFPUG. Boca Raton, FL: CRC Press, 2012.

24. Valerdi, R. “Convergence of Expert Opinion via the Wideband Delphi Method: An Application in Cost Estimation Models.” 21st Annual INCOSE International Symposium. Denver, CO, June 2011.

25. Valerdi, R. and Blackburn, C. "The Human Element of Decision Making in Systems Engineering: A Focus on Optimism." 19th Annual INCOSE International Symposium, Singapore, July 19-23, 2009.

# REPORT DOCUMENTATION PAGE

*Form Approved*
*OMB No. 0704-0188*



| 1. AGENCY USE ONLY | 2. REPORT DATE | 3. REPORT TYPE AND DATES COVERED |
|---|---|---|
| (Leave Blank) | March 2013 | Final |

| 4. TITLE AND SUBTITLE | 5. FUNDING NUMBERS |
|---|---|
| Quantifying Uncertainty in Expert Judgment: Initial Results | FA8721-05-C-0003 |

6. AUTHOR(S)

Dennis R. Goldenson, Robert W. Stoddard II

| 7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) | 8. PERFORMING ORGANIZATION REPORT NUMBER |
|---|---|
| Software Engineering Institute<br>Carnegie Mellon University<br>Pittsburgh, PA 15213 | CMU/SEI-2013-TR-001 |

| 9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) | 10. SPONSORING/MONITORING AGENCY REPORT NUMBER |
|---|---|
| AFLCMC/PZE/Hanscom<br>Enterprise Acquisition Division<br>20 Schilling Circle<br>Building 1305<br>Hanscom AFB, MA 01731-2116 | ESC-TR-2013-001 |

11. SUPPLEMENTARY NOTES

| 12A DISTRIBUTION/AVAILABILITY STATEMENT | 12B DISTRIBUTION CODE |
|---|---|
| Unclassified/Unlimited, DTIC, NTIS | |

13. ABSTRACT (MAXIMUM 200 WORDS)

The work described in this report, part of a larger SEI research effort on Quantifying Uncertainty in Early Lifecycle Cost Estimation (QUELCE), aims to develop and validate methods for calibrating expert judgment. Reliable expert judgment is crucial across the program acquisition lifecycle for cost estimation, and perhaps most critically for tasks related to risk analysis and program management. This research is based on three field studies that compare and validate training techniques aimed at improving the participants' skills to enable more realistic judgments commensurate with their knowledge.

Most of the study participants completed three batteries of software engineering domain-specific test questions. Some participants completed four batteries of questions about a variety of general knowledge topics for purposes of comparison. Results from both sets of questions showed improvement in the participants' recognition of their true uncertainty. The domain-specific training was accompanied by notable improvements in the relative accuracy of the participants' answers when more contextual information to the questions was given along with "reference points" about similar software systems. Moreover, the additional contextual information in the domain-specific training helped the participants improve the accuracy of their judgments while also reducing their uncertainty in making those judgments.

| 14. SUBJECT TERMS | 15. NUMBER OF PAGES |
|---|---|
| QUELCE, cost estimation, calibration, expert judgment, estimation modeling | 63 |

16. PRICE CODE

| 17. SECURITY CLASSIFICATION OF REPORT | 18. SECURITY CLASSIFICATION OF THIS PAGE | 19. SECURITY CLASSIFICATION OF ABSTRACT | 20. LIMITATION OF ABSTRACT |
|---|---|---|---|
| Unclassified | Unclassified | Unclassified | UL |