Effect of Level of Problem Description on Problem Discovery Rate: Two Case Studies

TR 29.3604
December 18, 2002

James R. Lewis

IBM Voice Systems

Boca Raton, Florida

Abstract

The primary purpose of this study was to investigate the effect of changing the level of description of usability problems on the estimate of the problem discovery rate. A secondary purpose was to describe a method for using the problem discovery rate to estimate the number of problems (rather than just the percentage) remaining available for discovery given the constraints associated with a particular participant population, application, and set of tasks. The results indicated that the level of problem description influences estimates of problem discovery rates. Furthermore, the direction of influence seemed to be predictable, with higher levels of description producing higher estimates of *p*. Practitioners need a level of description that flows easily into recommendations for redesigning products. Any other level of description places severe limitations on the practical utility of a usability study. On the other hand, to keep usability studies as efficient as possible by maximizing adjusted values for *p*, practitioners also need to seek a level of description that takes advantage of common patterns in observed usability problems. Managing this tradeoff is only one of the challenges of usability evaluation, but it is an important one.

ITIRC Keywords

Usability problems
Level of description
Problem discovery rate
Estimating the number of discoverable usability problems
Sample size estimation
Sample size adequacy

Contents

Introduction

Motivation

A current area of usability research is to develop an understanding of the fundamental properties of usability problems. For example, one important outstanding issue is the development of a definition of what constitutes a 'real' usability problem with which a broad base of usability scientists and practitioners can agree (Cockton & Lavery, 1999; Connell & Hammond, 1999; Lavery, Cockton, & Atkinson, 1997; Lee, 1998; Virzi, Sokolov, & Karis, 1996).

Included in this issue is the appropriate level of description of usability problems. If the purpose of a usability study is to discover and fix problems during system development, than it might be necessary to describe the problems at a fairly low level to ensure a specific enough description to guide efforts to redesign the system. Alternatively, if the purpose is to map problems onto a theoretical or heuristic framework (such as that of Nielsen, 1994, or Limin, Salvendy, & Turley, 2002), it might be necessary to describe the problems at a higher level.

Another area of research in the properties of usability problems is the estimation of the problem discovery rate *p* (Lewis, 1992, 1994, 2001; Nielsen & Landauer, 1993; Virzi, 1990, 1992). This estimate is useful in planning sample size requirements for usability studies and, after completing a study, in assessing the adequacy of the sample size.

The primary purpose of the current study was to investigate the effect of changing the level of description of usability problems on the estimate of the problem discovery rate. A secondary purpose was to describe a method for using the problem discovery rate to estimate the number of problems (rather than just the percentage) remaining available for discovery given the constraints associated with a particular participant population, application, and set of tasks.

Sample size estimation for problem-discovery studies

Estimating sample sizes for studies that have the primary purpose of discovering the problems in an interface depends on having an estimate of *p*, the average likelihood of problem occurrence (which is also an estimate of the problem discovery rate). This estimate can come from previous studies using the same method and similar system under evaluation, or can come from a pilot study. For standard scenario-based usability studies, the literature contains large-sample examples with *p* ranging from .16 to .42 (Lewis, 1994). For heuristic evaluations, the reported value of *p* from large-sample studies ranges from .22 to .60 (Nielsen and Molich, 1990).

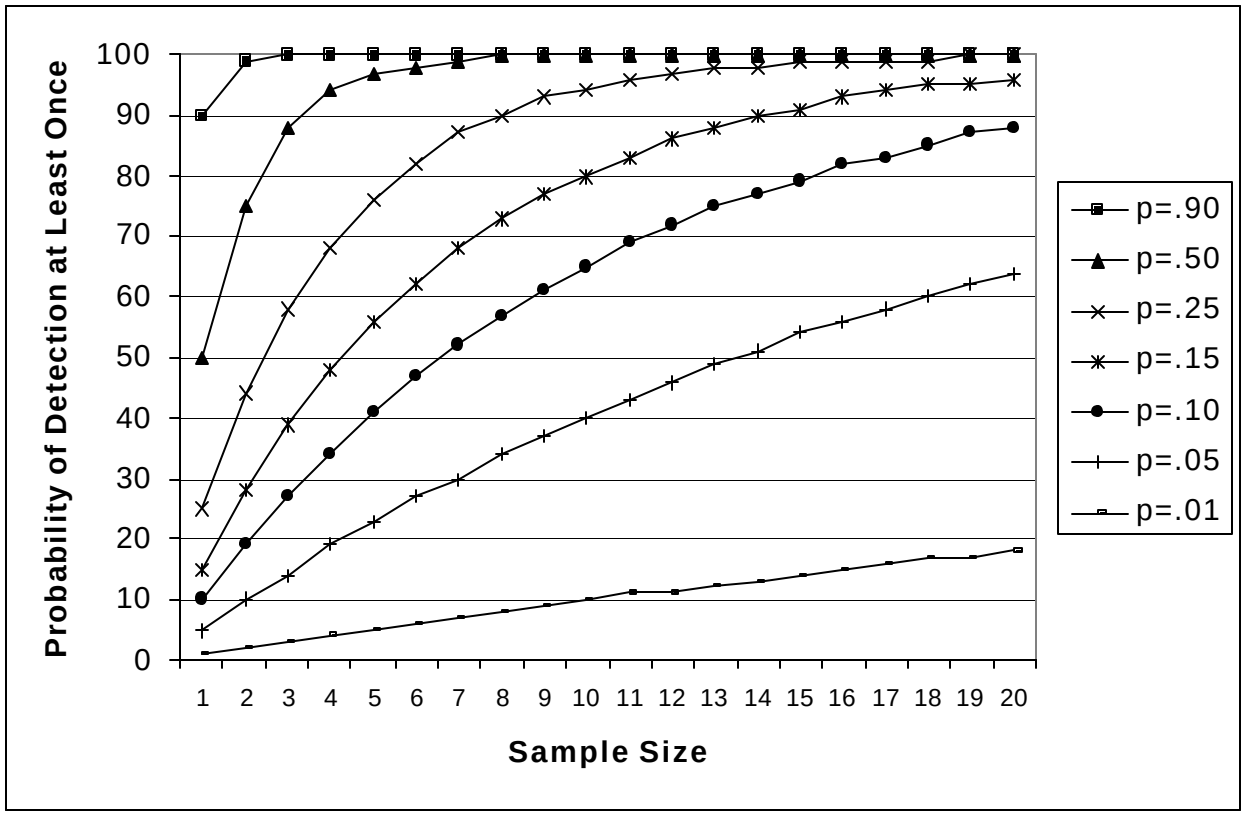
When estimating *p* from a small sample (say, fewer than 20 participants), it is important to adjust its estimated value because small-sample estimates of *p* have a bias that results in potentially substantial overestimation (Hertzum and Jacobsen, 2001). A series of recent Monte Carlo experiments (Lewis, 2001) have demonstrated that a formula combining Good-Turing discounting with a normalization procedure provides a very accurate adjustment of initial estimates of *p*, even when the sample size for that initial estimate has as few as two participants. This formula for the adjustment of *p* is:

$$[1]\ adjp = \tfrac{1}{2}[(estp - 1/n)(1 - 1/n)] + \tfrac{1}{2}[estp/(1+GTadj)]$$

where *GTadj* is the Good-Turing adjustment to probability space (which is the proportion of the number of problems that occurred once divided by the total number of different problems). The *estp*/(1+*GTadj*) component in the equation produces the Good-Turing adjusted estimate of *p* by dividing the observed, unadjusted estimate of *p* (*estp*) by the Good-Turing adjustment to probability space. The (*estp* – 1/*n*)(1 – 1/*n*) component in the equation produces the normalized estimate of *p* from the observed, unadjusted estimate of *p* and *n* (the sample size used to estimate *p*). The reason for averaging these two different estimates is that the Good-Turing estimator consistently tends to overestimate the true value of *p*, and the normalization consistently tends to underestimate it (Lewis, 2001).

Once you have an adjusted estimate for *p*, you can use the formula $1-(1-p)^n$ (derived from the binomial probability formula, Lewis, 1982, 1994, or, alternatively, from the Poisson probability formula, Nielsen & Landauer, 1993) with various values of *n* (say, from 1 to 20) to generate the curve of diminishing returns expected as a function of sample size (illustrated in Figure 1 for a range of values of *p*).

Figure 1. Predicted Discovery as a Function of Estimated Problem Discovery Rate



Usability practitioners can use these curves to estimate sample size requirements for a usability study, typically by selecting some goal for percentage of problem discovery (say, 90%) and checking to see at what sample size for a given problem discovery rate the curve crosses 90%. Practitioners who must use a given sample size can use a variation of this technique to assess the effectiveness of the sample size by determining the percentage of problems discovered with the given sample size for the adjusted estimate of *p*.

Estimating the Number of Remaining Usability Problems

Because a practitioner knows the number of problems discovered for a given sample size in any specific usability study (given that practitioner's criteria for the identification and classification of usability problems), it is possible to use the estimates for the percentage of discovered problems to calculate the number of remaining problems and their likely pattern of discovery.

For example, suppose a practitioner has collected data from three participants, observed 12 distinct usability problems, and determined that the adjusted value of *p* was .25. From Figure 1, the percentage of problems discovered with three participants when *p*=.25 should be about 58%. If 12 problems are 58% of the total available for discovery, then the total available for discovery is 12/.58, or 21. Again referring to Figure 1, to achieve a problem-discovery goal of 90% (about 19 problems), the practitioner would need to collect data from a total of 8 participants (in other words, would need to run 5 more). Table 1 shows the estimated pattern of discovery of the additional usability problems. Note that for participants 5-8 the expected rate of discovery is one new usability problem per additional participant.

Table 1. Hypothetical Estimated Pattern of Discovery of Additional Usability Problems

n	cum(p)	Problems
0	0.00	0
1	0.25	5
2	0.44	9
3	0.58	12
4	0.68	14
5	0.76	16
6	0.82	17
7	0.87	18
8	0.90	19

Case Study 1

Introduction

The prototype in the first case study simulated a system that had Weather, News, and E-mail/Calendar applications. The user interface style for all applications was directed dialog.

Method

Participants. Three IBM employees participated in this study. Two of the participants were male (1 < 40 years of age; 1 > 40 years) and one was female (< 40 years of age). All three were experienced e-mail users.

Apparatus and Materials. Participants completed five tasks (with tasks presented in the same order for all participants).

Procedure. Each participant received a brief description of the specific system setup and the capabilities of the applications. Each participant read through each of the five tasks, asked for any necessary clarifications, and indicated to the experimenter when he or she was ready to begin the tasks. Participants received instruction to attempt to complete all tasks in a single phone call. Table 2 gives a brief description of each task.

Table 2. Brief Task Descriptions for Case Study 1

Task #	Task Name	Description
1	Authentication/Weather	Listen to the weather for Miami
2	News	Listen to a news headline about a NASCAR race car driver
3	Find message	Find a specific message and listen to it
4	Review Appointments	Find out the time for your first appointment tomorrow
5	Reminder	Set a reminder to buy milk

Problems Discovered during Usability Testing (Low Level of Description)

Usability testing revealed 6 usability problems (using a low-level of description), summarized in Table 3.

Table 3. Summary of Observed Usability Problems in Study 1 (Low-Level Description)

Problem Description	Part 1	Part 2	Part 3	Frequency
1. Task 2: False stop	0	1	1	67
2. Task 4: Nomatch – back on task with Help 1	0	1	0	33
3. Task 5: Misleading prosody on prompt	0	1	0	33
4. Task 1: Nomatch – back on task with Help 1	0	0	1	33
5. Task 4: Exploring – searching for good option	0	0	1	33
6. Task 5: Nomatch – back on task with Help 1	0	0	1	33

Note: *Frequency* = Percentage of participants who experienced the problem.

The six observed problems had low frequency except for accidental stopping of system playback of a news story in Task 2. The three nomatch events (failures to match what the user said with anything in the active grammar) occurred in three different tasks. In each case, the first level of help that played in response to the nomatch event put the participant back on task.

Adequacy of Sample Size for Low Level of Description

With five of the six observed problems occurring only once during the study, it appeared that the problem space defined by this test situation (participants, application, and tasks) was fairly sparse, suggesting that there would be little benefit (relative to cost) gained by continuing to search for additional problems by running more participants. Here are the steps taken to evaluate the situation quantitatively, using the procedure published in Lewis (2001).

First, you need to calculate the initial estimated value of p (*estp*). One way to do this is to divide the number of observed problems by the number of known opportunities to observe problems. As shown in Table 3 above, for this case the value is 7/18[1], or .39.

Because the initial estimate of p from a small sample will have an inflation bias (Lewis, 2001), the second step is to adjust it. Using the formula given in [1], the adjusted value of p (*adj-p*) is .125 (less than half the initial estimate). Here are the computational steps taken to arrive at this adjusted value:

- The formula given in [1] was $adjp = \frac{1}{2}[(estp - 1/n)(1 - 1/n)] + \frac{1}{2}[estp/(1+GTadj)]$.
- As described above, *estp* is .39 and n is 3.
- *GTadj* is the number of problem types that occurred once (5 in this example) divided by the total number of problem types (6 in this example), or .83.
- So, $adjp = \frac{1}{2}[(.39 - 1/3)(1 - 1/3)] + \frac{1}{2}[.39 /(1+.83)]$, = $\frac{1}{2}[(.06)(.67)] + \frac{1}{2}[.39 /(1.83)]$, = $\frac{1}{2}[(.04)] + \frac{1}{2}[.21)]$, = .02 + .105, = .125.

Projecting to a sample size of 3 with the cumulative binomial probability formula $(1-(1-p)^n)$ with p=.125 and n=3, the probable proportion of discovered problems is about .33. Therefore, the total number of problems available for discovery in this problem space (at the given level of problem definition) is about 18 (6/.33). Table 4 shows the most likely pattern of problem discovery given these conditions. Figures 2 and 3 illustrate the patterns for proportion of problems discovered and numbers of problems discovered, respectively.

For participant 4, the expectation is the discovery of two more new problems. For the next five participants (5-9), the expectation is the discovery of one additional new problem per participant. After that, the expected discovery of new problems becomes even less frequent. To discover 90% of the problems available for discovery would take about 17 participants (an additional 14 participants) – to find 10 more problems. This suggests that 90% might not be a

[1] From Table 3, you can see that there were 7 observed problems. The total number of opportunities is the number of cells in the problem by participant matrix which, in this case, is 3 x 6, or 18.

reasonable problem discovery goal for this specific situation due to the amount of resource required to uncover each new usability problem. Whether to continue testing in this problem space or to switch to another problem space (for example, switching to different types of participants or different tasks) is a matter of practitioner judgement, aided by these calculations.

Table 4. Likely Pattern of Problem Discovery for Case Study 1 (Low-Level Description)

Sample Size (*n*)	Cumulative Proportion of Discovery	Number of Discovered Problems	Additional Participants Needed		Sample Size (*n*)	Cumulative Proportion of Discovery	Number of Discovered Problems	Additional Participants Needed
0	0.00	0	na		12	0.80	14	9
1	0.12	2	na		13	0.82	15	10
2	0.23	4	na		14	0.84	15	11
3	0.33	6	na		15	0.86	16	12
4	0.41	8	1		16	0.88	16	13
5	0.49	9	2		17	0.90	16	14
6	0.55	10	3		18	0.91	17	15
7	0.61	11	4		19	0.92	17	16
8	0.66	12	5		20	0.93	17	17
9	0.70	13	6		21	0.94	17	18
10	0.74	13	7		22	0.95	17	19
11	0.77	14	8					

Figure 2. Projected Proportion Problem Discovery for Case Study 1 (Low-Level Description)

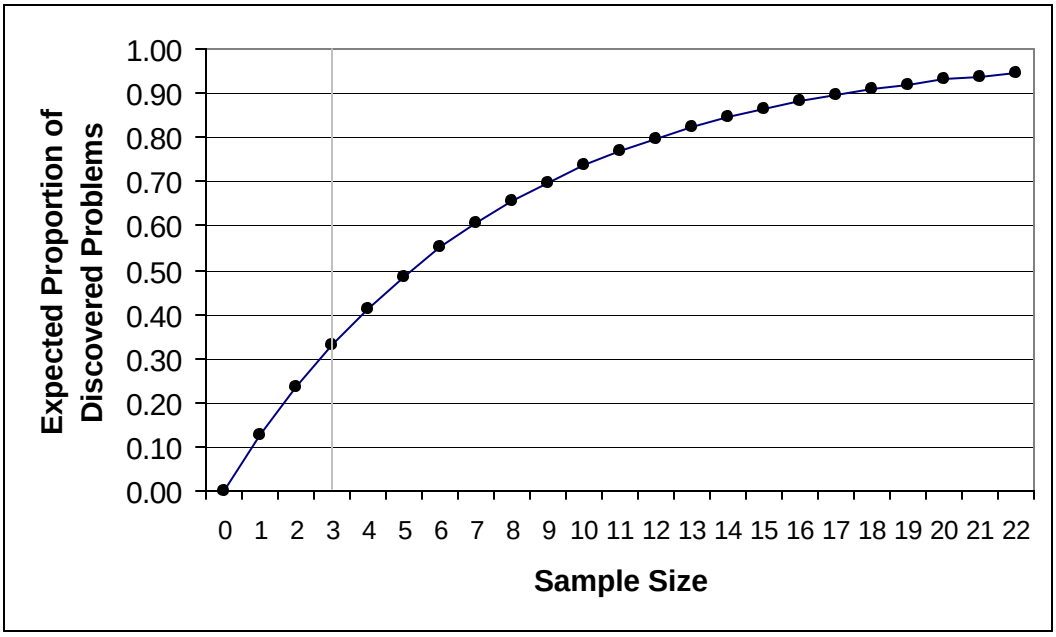
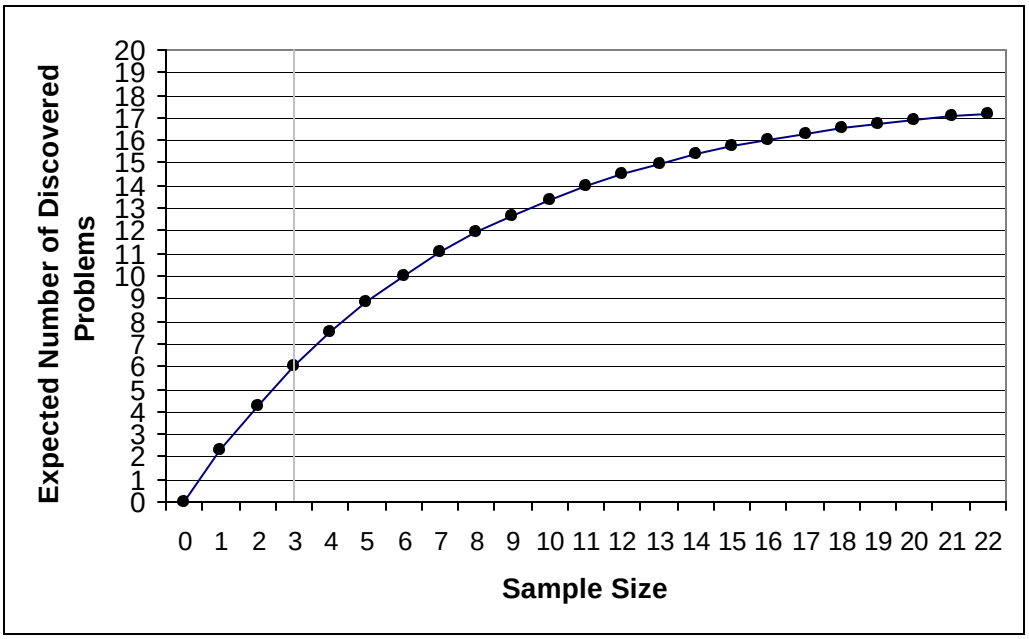


Figure 3. Projected Numbers of Discovered Problems for Case Study 1 (Low-Level Description)



Reanalysis at a Higher Level of Description

One way to shift to a higher level of description for the usability problems is to collapse problem types over task scenarios (specifically, combining the data for the three occurrences of the "Nomatch – back on task with Help 1" problem into a single category), as shown in Table 5. At this level of description, the usability study uncovered four types of problems, with the distribution across participants as shown in the table.

Table 5. Summary of Observed Usability Problems in Study 1 (High-Level Description)

Problem Description	Part 1	Part 2	Part 3	Frequency
1. False stop (Imp=3)	0	1	1	67
2. Nomatch - Help 1 (Imp=3)	0	1	1	67
3. Bad prosody on prompt (Imp=4)	0	1	0	33
4. Exploring (Sev=4)	0	0	1	33

Note: *Frequency* = Percentage of participants who experienced the problem.

Adequacy of Sample Size for High Level of Description

From Table 5, the initial estimate of p (*estp*) was .50, and the adjusted estimate (*adjp*) was .22. Given these values, a sample size of 3 should uncover about 53% of the problems available for discovery (which, with four problems at this level of description observed, suggests that three or four problems remained unseen).

Table 6 shows the most likely pattern of problem discovery given these conditions. Figures 4 and 5 illustrate the patterns for proportion of problems discovered and numbers of problems

discovered, respectively. The data show that the expected rate of discovery of new problems is less than one per participant for any additional participants.

Table 6. Likely Pattern of Problem Discovery for Case Study 1 (High-Level Description)

Sample Size (*n*)	Cumulative Proportion of Discovery	Number of Discovered Problems	Additional Participants Needed		Sample Size (*n*)	Cumulative Proportion of Discovery	Number of Discovered Problems	Additional Participants Needed
0	0.00	0	na		5	0.72	5	2
1	0.22	2	na		6	0.78	6	3
2	0.40	3	na		7	0.83	6	4
3	0.53	4	na		8	0.87	7	5
4	0.63	5	1		9	0.90	7	6

Figure 4. Projected Proportion Problem Discovery for Case Study 1 (High-Level Description)

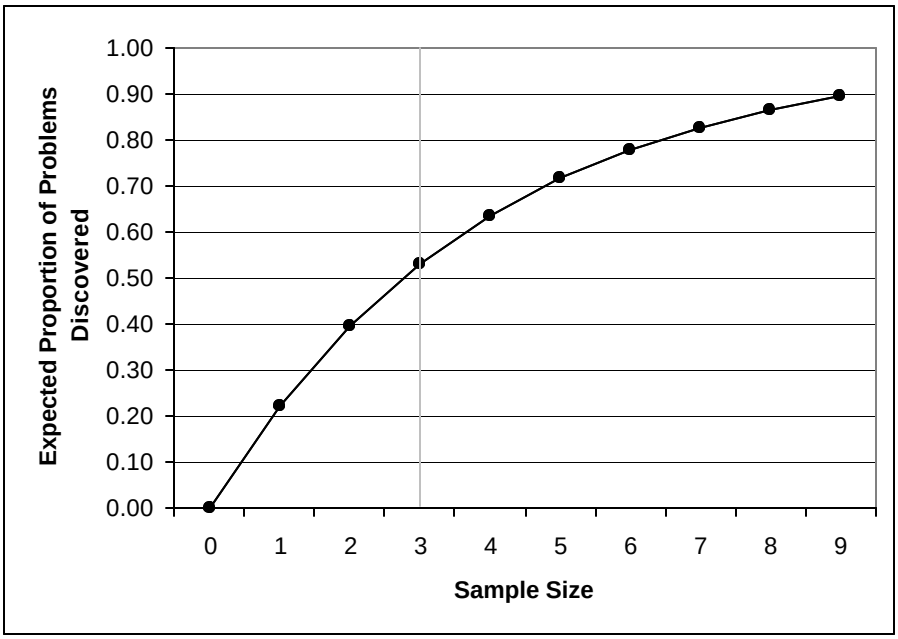
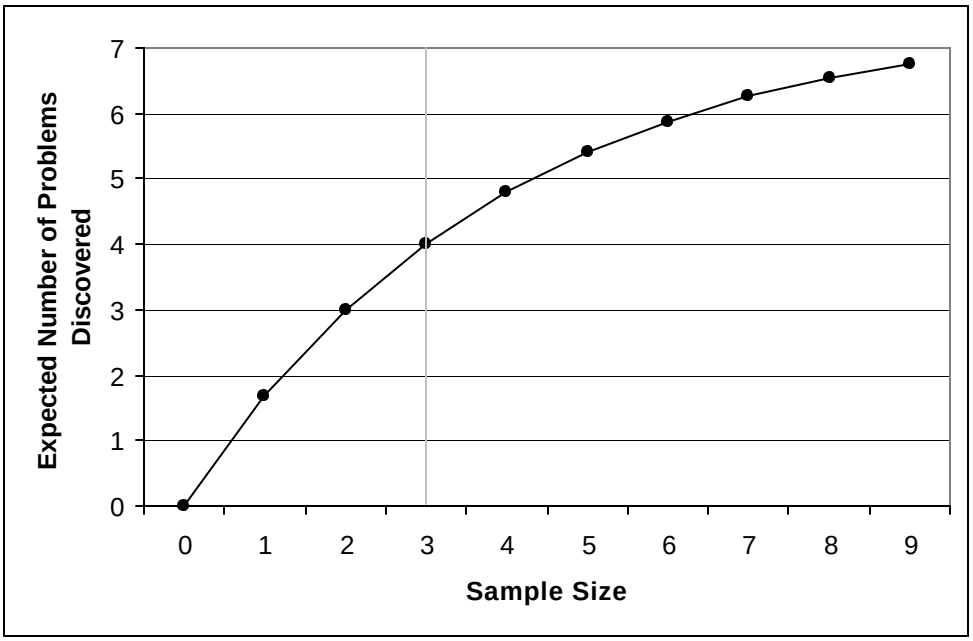


Figure 5. Projected Numbers of Discovered Problems for Case Study 1 (High-Level Description)



Case Study 2

Introduction

The prototype in the second case study also had Weather, News, and Lotus Notes applications, but had design changes intended to eliminate or reduce the usability problems observed for the first prototype. It also had a natural command grammar for the e-mail/calendar application, as well as a number of additional design changes.

Method

Participants. The study had seven participants (one IBM employee and six people hired from a temporary employment agency). All participants had experience with web browsing and e-mail, but generally to a lesser extent than the participants who worked with the first prototype.

Apparatus, Materials, and Procedure. The apparatus, materials, and procedure were identical to those of Case Study 1, with the following exceptions:

- Updated prototype code to implement recommendations from Study 1
- Updated prototype code to implement natural commands grammar for e-mail/calendar application
- Changed Scenario 3 to include the tasks of determining the date and time when the note was sent and creating a reply to the note (making this task considerably more complex than in the previous study)

Problems Discovered during Usability Testing (Low Level of Description)

Usability testing revealed 33 usability problems (using a low-level of description), summarized in Table 8 below.

Adequacy of Sample Size for Low Level of Description

From Table 8, the initial estimate of p (*estp*) was .27, and the adjusted estimate (*adjp*) was .15. Given these values, a sample size of 7 should uncover about 68% of the problems available for discovery (which, given 33 observed problems at this level of description, suggests that there were about 49 problems available for discovery in this problem space, with about 16 problems remaining undiscovered).

Table 8. Summary of Observed Usability Problems in Study 2 (Low-Level Description)

Problem Description	Part 1	Part 2	Part 3	Part 4	Part 5	Part 6	Part 7	Freq
1. Task 1 -- Encountered help level 1	0	0	0	1	1	1	1	57
2. Task 1 -- Attempted natural command	0	0	0	0	1	0	0	14
3. Task 2 -- Encountered help level 1	0	0	0	0	1	0	1	29
4. Task 2 -- Accidentally exited	0	0	1	0	0	0	0	14
5. Task 3 -- Breath noise misrecoed	1	0	0	0	0	0	0	14
6. Task 3 -- Encountered help level 1	1	1	1	1	0	1	1	86
7. Task 3 -- Confused by Reply to All?	0	1	1	0	0	0	1	43
8. Task 3 -- False prompt stop	0	0	1	0	0	0	0	14
9. Task 3 -- Natural command inactive	0	0	1	0	0	0	0	14
10. Task 3 -- Encountered help level 2	0	0	1	0	0	1	0	29
11. Task 3 -- Encountered help level 3	0	0	1	0	0	1	0	29
12. Task 3 -- Navigation error	0	0	1	0	0	1	0	29
13. Task 3 -- Accidentally exited	0	0	0	1	0	0	1	29
14. Task 3 -- Natural command nomatch	0	0	0	0	0	0	1	14
15. Task 4 -- Natural command interrupted	1	0	0	0	1	0	0	29
16. Task 4 -- Experienced help level 1	1	0	0	1	0	0	1	43
17. Task 4 -- Experienced help level 2	0	0	0	1	0	0	0	14
18. Task 4 -- Experienced help level 3	0	0	0	1	0	0	0	14
19. Task 4 -- Natural command inactive	1	0	0	1	1	0	0	43
20. Task 4 -- Said application name to restart it	0	0	0	1	0	0	0	14
21. Task 4 -- False prompt stop	0	0	0	1	0	1	0	29
22. Task 4 -- Fell to directed dialog	1	0	0	1	1	0	1	57
23. Task 4 -- Misrecognized date	0	0	0	1	0	0	0	14
24. Task 5 -- Misrecognized date	1	0	0	0	0	0	0	14
25. Task 5 -- Natural command inactive	1	0	0	0	0	0	0	14
26. Task 5 -- Fell to directed dialog	1	1	0	0	0	0	0	29
27. Task 5 -- Experienced help level 1	0	1	0	1	0	1	0	43
28. Task 5 -- Got play instead of create	0	0	1	0	1	0	0	29
29. Task 5 -- Experienced help level 2	0	0	0	1	0	1	0	29
30. Task 5 -- Incorrect natural command	0	0	0	1	0	1	0	29
31. Task 5 -- False prompt stop	0	0	0	1	0	0	0	14
32. Task 5 -- Pause after example helps too short	0	0	0	0	1	0	0	14
33. Task 5 -- Confusion about recording	0	0	0	0	0	0	1	14

Note: *Frequency* = Percentage of participants who experienced the problem.

Table 9 shows the most likely pattern of problem discovery given these conditions. Figures 4 and 5 illustrate the patterns for proportion of problems discovered and numbers of problems discovered, respectively.

Table 9. Likely Pattern of Problem Discovery for Case Study 2 (Low-Level Description)

Sample Size (*n*)	Cumulative Proportion of Discovery	Number of Discovered Problems	Additional Participants Needed		Sample Size (*n*)	Cumulative Proportion of Discovery	Number of Discovered Problems	Additional Participants Needed
0	0.00	0	na		10	0.80	39	3
1	0.15	7	na		11	0.83	40	4
2	0.28	13	na		12	0.86	42	5
3	0.38	19	na		13	0.88	43	6
4	0.48	23	na		14	0.90	44	7
5	0.55	27	na		15	0.91	44	8
6	0.62	30	na		16	0.92	45	9
7	0.68	33	na		17	0.94	46	10
8	0.72	35	1		18	0.95	46	11
9	0.77	37	2		19	0.95	46	12

Figure 6. Projected Proportion Problem Discovery for Case Study 2 (Low-Level Description)

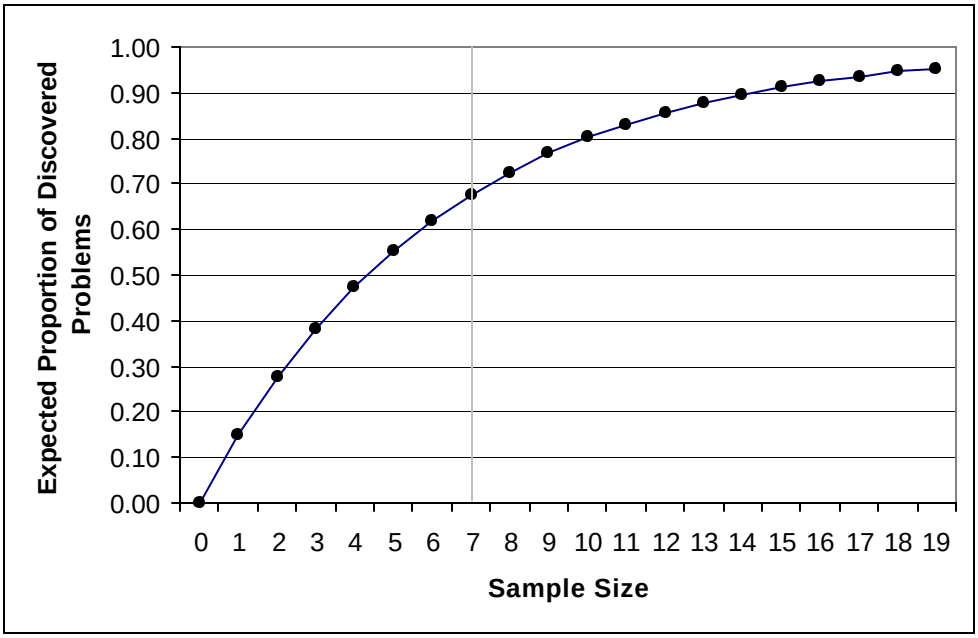
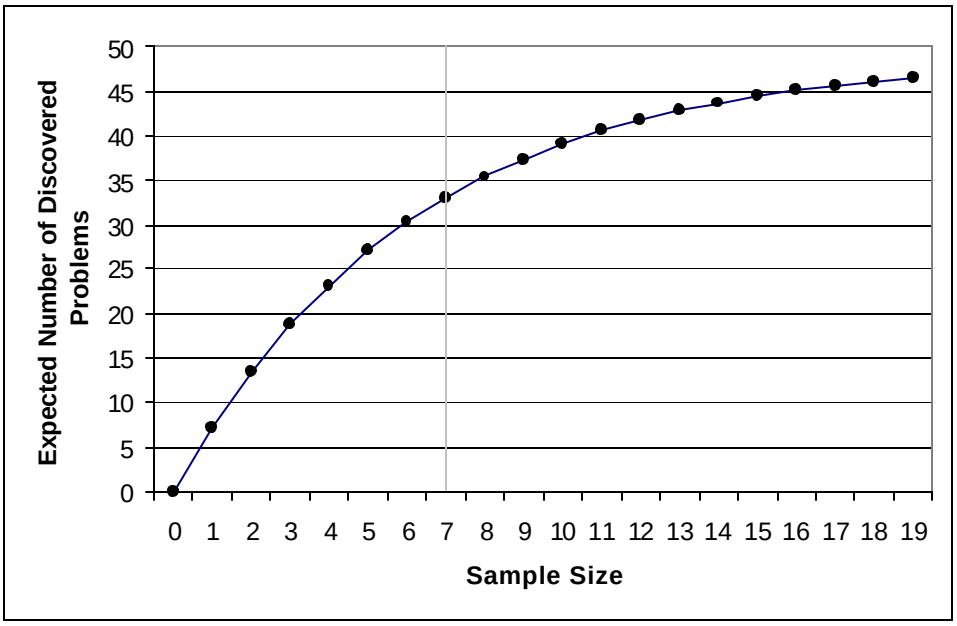


Figure 7. Projected Numbers of Discovered Problems for Case Study 2 (Low-Level Description)



Reanalysis at a Higher Level of Description

As in Case Study 1, I collapsed problem types over task scenarios to move to a higher level of description. At this level of description, the usability study uncovered 18 types of problems, with the distribution across participants as shown in Table 10.

Table 10. Summary of Observed Usability Problems in Study 2 (High-Level Description)

Problem Description	Part 1	Part 2	Part 3	Part 4	Part 5	Part 6	Part 7	Freq
1. Accidentally exited	0	0	1	1	0	0	1	43
2. Breath noise misrecoed	1	0	0	0	0	0	0	14
3. Confused by Reply to All?	0	1	1	0	0	0	1	43
4. Confusion about recording	0	0	0	0	0	0	1	14
5. Encountered help level 1	1	1	1	1	1	1	1	100
6. Encountered help level 2	0	0	1	1	0	1	0	43
7. Encountered help level 3	0	0	1	1	0	1	0	43
8. False prompt stop	0	0	1	1	0	1	0	43
9. Fell to directed dialog	1	1	0	1	1	0	1	71
10. Got play instead of create reminder	0	0	1	0	1	0	0	29
11. Incorrect natural command	0	0	0	1	0	1	0	29
12. Misrecognized date	1	0	0	1	0	0	0	29
13. Natural command inactive	1	0	1	1	1	0	0	57
14. Natural command interrupted	1	0	0	0	1	0	0	29
15. Natural command OOG	0	0	0	0	0	0	1	14
16. Navigation error	0	0	1	0	0	1	0	29
17. Pause after exp helps too short	0	0	0	0	1	0	0	14
18. Said application name to restart it	0	0	0	1	0	0	0	14

Note: *Freq* = Percentage of participants who experienced the problem.

Adequacy of Sample Size for High Level of Description

From Table 10, the initial estimate of *p* (*estp*) was .36, and the adjusted estimate (*adjp*) was .235. Given these values, a sample size of seven should have uncovered about 85% of the problems available for discovery (which, with 18 observed problems at this level of description, suggests that there were about 21 problems available for discovery, with 3 remaining undiscovered).

Table 11 shows the most likely pattern of problem discovery given these conditions. Figures 8 and 9 illustrate the patterns for proportion of problems discovered and numbers of problems discovered, respectively.

Table 11. Likely Pattern of Problem Discovery for Case Study 2 (High-Level Description)

Sample Size (*n*)	Cumulative Proportion of Discovery	Number of Discovered Problems	Additional Participants Needed		Sample Size (*n*)	Cumulative Proportion of Discovery	Number of Discovered Problems	Additional Participants Needed
0	0.00	0	na		6	0.80	17	na
1	0.24	5	na		7	0.85	18	na
2	0.41	9	na		8	0.88	19	1
3	0.55	12	na		9	0.91	19	2
4	0.66	14	na		10	0.93	20	3
5	0.74	16	na		11	0.95	20	4

Figure 8. Projected Proportion Problem Discovery for Case Study 2 (High-Level Description)

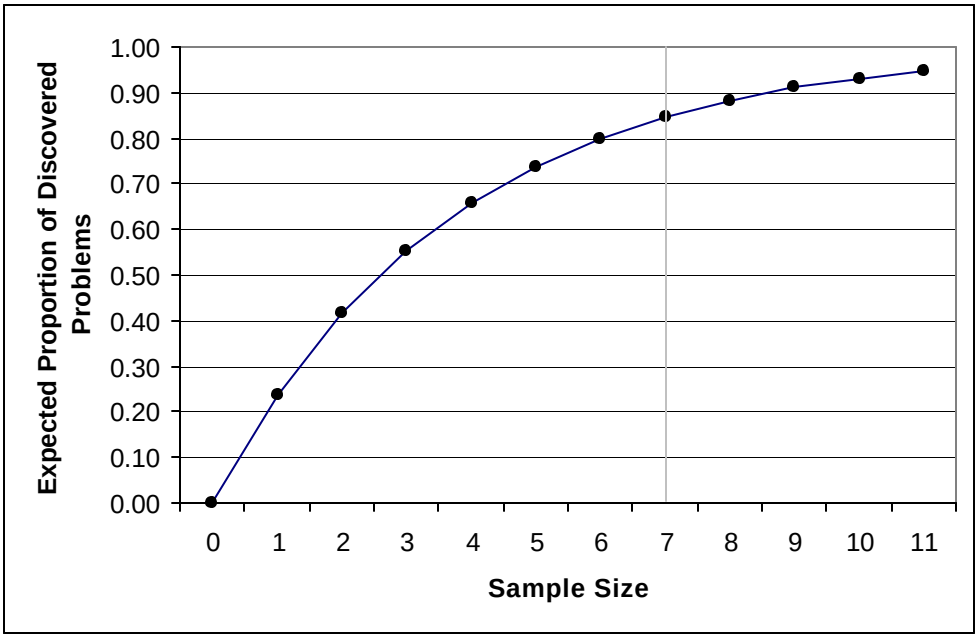
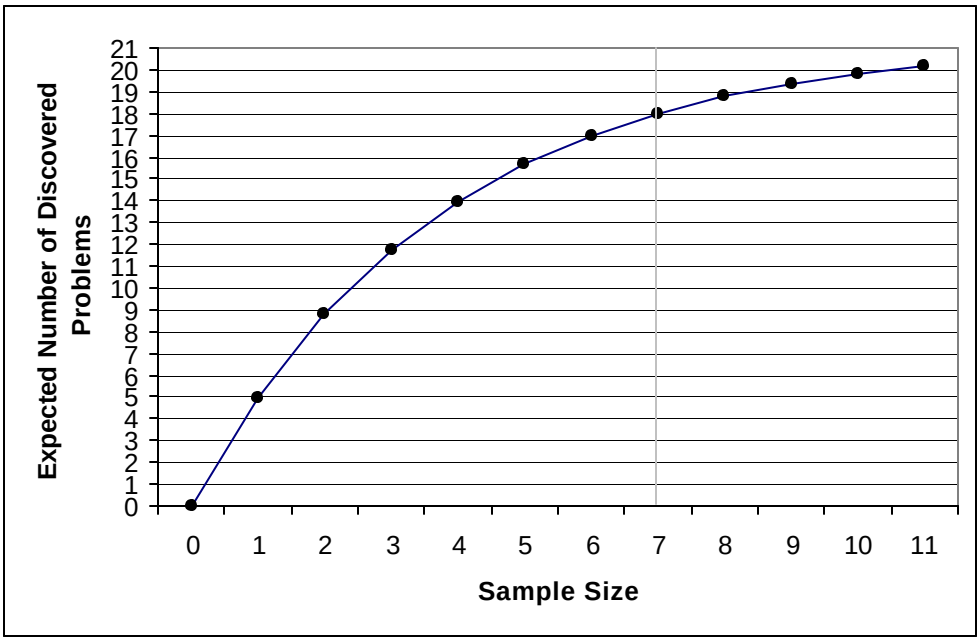


Figure 9. Projected Numbers of Discovered Problems for Case Study 2 (High-Level Description)



.

Discussion

These case studies indicate that the level of problem description influences estimates of problem discovery rates. Furthermore, the direction of influence seems to be predictable, with higher levels of description producing higher estimates of *p*. Such an outcome is perfectly reasonable.

Effect of the Highest Possible Level of Description

For example, consider the highest possible level of description, which is simply that a problem exists. This has the effect of collapsing all problems into a single row, as illustrated in Tables 12 and 13 for the data presented in these two case studies. For this general situation, it is very unlikely that the value of *GTadj* would be anything other than 0, removing any effect of Good-Turing discounting from the adjustment of *p*.

Table 12. Case Study 1 Data Given Highest Possible Level of Problem Description

Problem Description	Part 1	Part 2	Part 3	*estp*	*adjp*
Case Study 1 Problems	0	1	1	.67	.45

Table 13. Case Study 2 Data Given Highest Possible Level of Problem Description

Problem Description	Part 1	Part 2	Part 3	Part 4	Part 5	Part 6	Part 7	*estp*	*adjp*
Case Study 2 Problems	1	1	1	1	1	1	1	1.00	.87

Effect of the Lowest Possible Level of Description

The opposite extreme is to consider every observed problem to be completely unique. For Case Study 1, there was a total of 7 observed problems at the low level of problem description. In Case Study 2, the total at that level was 63.

If every problem is unique, then the value of *GTadj* is necessarily 1. The value of *estp* is the number of observed problems divided by the number of observed problems times the number of participants observed, which reduces to 1/*n*. For Case Study 1, the value of estp is .33, and for Case Study 2 it is .14. Because the numerator of the normalization portion of Equation [1] is *estp* – 1/*n*, and in this situation, *estp* = 1/*n*, this forces this part of the computation to be 0. Because the Good-Turing and normalization components are averaged in the equation, the effect is to make *adjp* equal to *estp*/4. Table 14 shows the resulting *p*-adjustment computations for each case study.

Table 14. Adjustments of p Given Lowest Possible Level of Problem Description

Computation	Case 1	Case2
GTadj:	1	1
.5(estp/(1+GTadj):	0.083	0.036
1/n:	0.333	0.143
.5(estp-1/n)(1-1/n):*	0	0
adjp:	0.08	0.04

Ranges of Adjusted Values of p as a Function of Level of Description

Thus, for the problems observed in Case Study 1, the adjusted estimate of *p* could range from .08 to .45, depending on the level of problem description. For Case Study 2, the value could range from .04 to .87. Table 15 shows the values of *adjp* for each of four situations for each case study: Minimum, Actual Low-Level, Actual High-Level, Maximum.

Table 15. Estimates of adjp for Four Situations

	Minimum	Actual Low-Level	Actual High-Level	Maximum
Case Study 1	0.08	0.125	0.22	0.45
Case Study 2	0.04	0.15	0.235	0.87

Implications for Practitioners

There are a few practical implications that follow from these data. Practitioners need a level of description that flows easily into recommendations for redesigning products. Any other level of description places severe limitations on the practical utility of a usability study. On the other hand, to keep usability studies as efficient as possible by maximizing adjusted values for *p*, practitioners need to seek a level of description that takes advantage of common patterns in observed usability problems. Managing this tradeoff is only one of the challenges of usability evaluation, but it is an important one.

References

Cockton, G., & Lavery, D. (1999). A framework for usability problem extraction. In *Human-Computer Interaction -- INTERACT '99* (pp. 344-352). Amsterdam: IOS Press.

Connell, I. W., & Hammond, N. V. (1999). Comparing usability evaluation principles with heuristics: Problem instances vs. problem types. *Human-Computer Interaction -- INTERACT '99* (pp. 621-629). Amsterdam: IOS Press.

Hertzum, M., & Jacobsen, N. (2001). The evaluator effect: A chilling fact about usability evaluation methods. *International Journal of Human-Computer Interaction, 13*, 421-443.

Lavery, D., Cockton, G., & Atkinson, M. P. (1997). Comparison of evaluation methods using structured usability problem reports. *Behaviour & Information Technology, 16*, 246-266.

Lee, W. O. (1998). Analysis of problems found in user testing using an approximate model of user action. In *People and Computers XIII: Proceedings of HCI '98* (pp. 23-35). Sheffield, UK: Springer-Verlag.

Lewis, J. R. (1982). Testing small-system customer set-up. In *Proceedings of the Human Factors Society 26th Annual Meeting* (pp. 718-720). Santa Monica, CA: Human Factors Society.

Lewis, J. R. (1994). Sample sizes for usability studies: Additional considerations. *Human Factors, 36*, 368-378.

Lewis, J. R. (2001). Evaluation of procedures for adjusting problem-discovery rates estimated from small samples. *International Journal of Human-Computer Interaction, 13*, 445-480.

Limin, F., Salvendy, G., & Turley, L. (2002). Effectiveness of user testing and heuristic evaluation as a function of performance classification. *Behaviour & Information Technology, 21*, 137-143.

Nielsen, J. (1994). Heuristic evaluation. In Nielsen, J., and Mack, R.L. (Eds.), *Usability Inspection Methods* (pp. 25-61). New York, NY: John Wiley.

Nielsen, J., & Landauer, T. K. (1993). A mathematical model of the finding of usability problems. In *Conference Proceedings on Human Factors in Computing Systems – CHI93* (pp. 206-213). New York, NY: Association for Computing Machinery.

Nielsen, J., & Molich, R. (1990). Heuristic evaluation of user interfaces. In *Conference Proceedings on Human Factors in Computing Systems – CHI90* (pp. 249-256). New York, NY: ACM.

Virzi, R. A. (1990). Streamlining the design process: Running fewer subjects. In *Proceedings of the Human Factors Society 34th Annual Meeting* (pp. 291-294). Santa Monica, CA: Human Factors Society.

Virzi, R. A. (1992). Refining the test phase of usability evaluation: How many subjects is enough? *Human Factors, 34*, 443-451.

Virzi, R. A., Sokolov, J. L., & Karis, D. (1996). Usability problem identification using both low- and high-fidelity prototypes. In *Conference on Human Factors in Computing Systems: CHI '96* (pp. 236-243). New York: Association for Computing Machinery.