Estimating sample size for usability testing

Alex Cazañas, Andre de San Miguel
University of Queensland
Brisbane, Australia
{alex.cazanagsgordon, andre.desanmiguel }
@uqconnect.edu.au

Esther Parra
Escuela Politécnica Nacional
Quito, Ecuador
esther.parra@epn.edu.ec

Abstract— One strategy used to discover user interface requirements is the conduct of usability testing. However, when conducting such testing one of the unknowns is how much testing should be done? Since conducting extensive testing is costly, reducing the number of tests can contribute greatly to successful resource management of a project. Even though a significant number of models have been proposed to estimate sample size in usability testing, there is still not consensus on the optimal sample size. Several studies claim that 3 to 5 users suffice to uncover 80% of problems in a software interface, however is this enough? This study uses data collected from the user testing of a web application in an attempt to verify these rules of thumb, collectively known as the “magic number five”. Analysis of the empirical data shows that the 5-user assumption significantly underestimates the number of users required to achieve reasonable levels of problem discovery.

Keywords— problem discovery; sample size; usability testing; human factor; computing system

I. INTRODUCTION

The degree by which a software product meets user expectations in terms of its ease of use, effectiveness and efficiency, reflects the accomplishment of the intended usability goals of a software development project. Conducting usability testing is central to the realization of such goals because it provides an effective tool to ensure that defects affecting usability are detected before the release of a new product.

In all projects, efficient resource allocation is essential, particularly when the cost of resources is high. Since, increasing the amount of user testing sessions directly impacts the project cost, budget constraints will limit the ability to conduct exhaustive user testing. Consequently, from an economical point of view, it is important to ensure that the benefit gained by additional testing is greater than the incurred costs.

In general, usability testing comprises a wide variety of methods and applications. The standard for usability reports ANSI INCITS 354-2001 states that usability tests consist of three major elements: participants, tasks, and environments. In addition, at least one observer must monitor the process thoroughly [1]. As per its primary focus, usability testing can be summative, if the emphasis is on task measurements; or formative, when the motivation is problem discovery. This distinction plays a major role in selecting the method to estimate sample size in a usability test.

Determining the minimum number of participants that exposes most problems in a usability test is a problem that has generated a considerable amount of research and debate during the past two decades.

In the early 1990’s, Virzi [2], Nielsen and Landauer [3], and Lewis [4] were the first to publish studies on methods for estimating sample size for problem discovery testing. Based on statistical modelling and empirical data, these authors proposed methods to estimate the size of the minimum sample required to reach a target rate of problem discovery in testing usability of software interfaces. Furthermore, they made three outstanding claims:

1) Most problems are discovered by the first four to five participants.

2) The increment in problem discovery after five participants is minimal.

3) ROI of usability testing can be maximized by minimizing the sample size.

Since their publication, these claims also known as “4±1” or “magic number five” have generated a great deal of discussion in usability evaluation communities, so much so that at the Computer-Human Interaction conference in 2003, a panel was dedicated to discussing this matter [5].

This paper will focus on and review the number of users necessary for problem discovery on the interface for a web application. As part of this analysis the 4±1 model for estimating the sample size required will also be reviewed.

II. BACKGROUND

To determine how many users are required for usability testing, three studies were considered as candidates for the provision of a suitable base calculation template.

Virzi [2] used empirical data from three experiments and Monte Carlo simulation to conclude that problem discovery rate and the number of participants establish an asymptotic relationship. In the three experiments, trained usability practitioners observed that the number of discovered problems depends on the number of participants in each experiment. The probability of discovering a problem (problem discovery rate) was computed for each participant as the quotient between the number of uncovered problems in a single session, and the total of unique problems in all sessions. The relationship between the number of participants and the discovery rate was modelled with the cumulative binomial probability formula, as follows...
Proportion of unique problems found = 1-(1-p)^n  \hspace{1cm} (1)

Where p is the mean problem discovery rate, and n is the number of participants; p may be calculated across participants or problems.

Similarly, Nielsen and Landauer [3] used data derived from eleven usability tests, and a Poisson distribution to reach a similar model

Number of unique problems found = N(1-(1-p)^i)  \hspace{1cm} (2)

Where p is the problem discovery rate, N is the total number of problems, and i is the number of participants.

Both approaches determine the number of participants for a given p and an objective value of problem discovery rate or proportion of all problems to be found.

Lewis [4] applied the techniques used in [2] to empirical data from usability testing conducted on a piece of software for office applications. The findings of this study coincided with the results in [2]. Nevertheless, he encouraged caution with small-sample problem discovery estimation, recommending its application only “if the expected p is high, if the study will be iterative, and if undiscovered problems will not have dangerous or expensive outcomes” [4].

Nielsen and Landauer [3] reported that five users were enough to discover 75% of the problems when testing an interface, while Virzi [2] stated that four to five users are enough to determine 80% of problems in the interface under evaluation. Lewis [4] observed that five or four participants uncover more than 80% of problems, provided that the value of p is between 0.3 and 0.4.

The claims derived from the aforementioned studies have received plenty of attention from scholars and practitioners, in particular the so called, “magic number 5”. Several authors have challenged this claim regarding the soundness of modelling problem rate discovery with a single value for p. For instance, Woolrych and Cockton [6] contend that problems do not affect users uniformly, thus estimation based on problem frequency is misleading. Caulton [7] suggests that due to the heterogeneity of users, different types of users will discover different kinds of problems. Therefore, the model should incorporate a term that considers the number of user subgroups. Turner, Lewis, and Nielsen [8] responded to criticism of sample size formulae by providing a method to adjust the estimated average problem frequency. On the other hand, Hwang, and Salvendy [9] argue that 10±2 is a more general rule for optimal sample size than 4±1.

III. METHODOLOGY

Lewis [10] reviewed several calculation techniques and synthesized them into one technique he regarded as the most accurate, based on his empirical testing and literature review. As such, we will be using his calculation technique as the starting point for our analysis.

The steps required for this technique are as follows:

- Count unique and repeat problems identified by individual users.
- Calculate parameters and metrics from this data from which to make observations.

One of the components of the last step in this process is using the calculated parameters to determine the number of user tests or samples required to achieve a level of problem discovery. Finally, this number is used to answer the question of whether the number of user tests undertaken suffices to identify the required number of problems.

To review the consistency of the parameters calculated we will be calculating the parameters required and results from two different datasets.

A. Data sources

To obtain data required for the proposed analysis, two datasets were sourced from two independent surveys of the interface of a web application. In both surveys, participants were requested to identify problems with the usability of the interface. In total, 34 different respondents participated in both surveys with 17 testers in each round. The second survey was undertaken two weeks after the first.

To prepare the user testing data set for parameter estimation, two passes were made over each set. The first pass was used to identify unique problems, which were then catalogued and numbered. Then, the second pass counted which users identified which problem or problems.

This process then resulted in a grid structure which shows the problem identification count for each successive user. An example of this structure can be seen in Table I.

This process also allows the identification of “repeat” problems i.e. those problems which are identified by more than one tester.

B. Parameter estimation

One of the outcomes available using the methodology provided by Lewis [10], is the calculation of two parameters n and p_adj, where:

n is an estimate of how many testers are required to discover a given proportion of total problems, calculated using the formula in Virzi [2]:

\[ n = \log(1 - \text{Goal})/\log(1 - p_{adj}) \]  \hspace{1cm} (3)

and p_adj is the problem occurrence which is calculated using the formula given in [10]:

\[ p_{adj} = \frac{1}{2} \left[p_{est} - 1/n \right] + \frac{1}{2} \left[p_{est}/(1 + G T_{adj}) \right] \]  \hspace{1cm} (4)

where n is the sample size used to compute the initial estimate of p, and GT_adj is the GT adjustment to probability space, which is the result of dividing the number of problems that occurred once by the number of different problems.

The calculation of p and n can then be performed for each successive data point on the problem count grid data table.
TABLE I. Problem Count Grid from Data-Set#2

<table>
<thead>
<tr>
<th>Participant</th>
<th>Problem Id</th>
<th>Problem Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1 1 1 - - - - - - 2</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>- - - - - - - - - - 0</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>- - - - - - - - - - 0</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>- 1 - - - - - - - 1</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>1 1 - - - - - - - 2</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>1 1 - - - - - - - 2</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>1 - - - - - - 1 - - 2</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>- 1 - - - - - - - 2</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>1 1 - - - - - - - 2</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>1 - - - - 1 - - 1 3</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>1 - - 1 1 - - 1 1 5</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>1 1 - - - - - - - - 3</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>- 1 - - - - - - - - 1</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>1 - - - - - - - - - 1</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>- - - - - - - - - - 0</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>1 - - - - - - - - - 1</td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>- 1 - - - - - - - - 1</td>
<td></td>
</tr>
<tr>
<td>Problem count</td>
<td>9 9 1 1 3 1 2 2 28</td>
<td></td>
</tr>
</tbody>
</table>

IV. RESULTS

The number of users required to uncover a given percentage of usability problems can be calculated for each number of user tests. Consequently, the aggregated computation across all user tests in both testing rounds created curves that provide the number of user tests required to accomplish a desired goal of problem discovery. Figures 1 and 2 show the estimate of the sample size required to detect 80%, 90% and 99% of problems in the interface under study. From the results of the first user test set (Figure 1), it can be seen that the estimated (mean) sample size required for a target of 80% of problem discovery is 9, with a standard deviation of 6.7, minimum of 5 and maximum of 26.

In the second round of testing, it can be seen that the estimated (mean) sample size required for a target of 80% of problem discovery is 9, with a standard deviation of 3, minimum of 4 and maximum of 18.

From the computed scores of the sample size of the two data sets, it is evident, that on average the size of the required sample to obtain an 80% of problem rate discovery is significantly larger than the value predicted by the 4±1 models.

V. DISCUSSION

There are two quite significant outcomes from the analysis conducted. The first brings into question the rules of thumb proposed by Virzi [2], Nielsen and Landauer [3], and Lewis [4], in the estimation of the number of user tests required to obtain the desired level of problem discovery.

The second significant outcome is the impact on the sample size of users required for ever increasing levels of problem discovery. As can be seen from the results from both data-sets, the amount of extra testing required to reach a 99% problem discovery outcome is significantly higher than that of the 90% level. This points towards a higher marginal cost of each extra problem discovered, especially once past the 90% threshold.

The analysis also confirms the commentary by Lewis [4] on the variability problems associated with using small samples to estimate the number of samples required for the required level of problem discovery.

Even though the goal of Lewis’s adjustments [10] to the p value were to designed improve the calculation of the sample size required for smaller data-sets, the smaller user-test end of the scale for both data-sets showed quite a lot of variability between the test sets, however on both charts the results level out after about eight or nine user tests, thus providing a level of comfort in the suggested sample size results.

This eventual levelling out of results suggests a way to use this type of analysis in practice. If the percentage of problem discovery required is known, after each user test this analysis can be run to determine if more tests are required. For example, assuming that 90% problem discovery is required, if this analysis had been done with Data-Set#1 then it would have been discovered that somewhere between 30 and 40 tests were required, thus pushing the testing past the actual 17 user tests.
Conversely, testing on Data-Set#2 could have stopped earlier as only 12 or 13 tests were required to hit the 90% problem discovery threshold.

In both scenarios, the question remains as to the “levelling out” point. However, this could be calculated using traditional statistical techniques.

While the levels of the rules of thumb are questionable, given the results of our study, the variability measures are reasonably accurate. For Data-Set#1 once the curve had settled at the 80% and 90% problem discovery levels, the standard deviation of the results curve was 2-3, comparing favorably with the ±2 value estimated by Hwang and Salvendy [8]. For Data-Set#2 the standard deviation is 0.8-1.0, which compares favorably with the ±1 in the 4±1 rule.

VI. CONCLUSIONS AND FUTURE WORK

Several studies [2], [3], [4], [7], [8], [9] suggest that when conducting user testing it is sufficient to use rules of thumb such as 4±1 and 10±2 to estimate the number of users required.

In our study, we discovered that the use of such rules of thumb would have significantly underestimated the actual number of users required to achieve reasonable levels of problem discovery. In the two data-sets studied, the number of users required to achieve 90% problem discovery were 12-13 and 30-40 respectively. This coincides with findings in [5], [6], [11], [12], and [13]. Furthermore, using small sample sizes can also be problematic as this produces large variability in testing results which cannot be fully adjusted for.

Since the potential costs of achieving problem discovery at the 99% level are significantly higher than that of achieving the 90% level, the use of these levels should be carefully considered unless the development is for applications for which the cost of problems is quite high.

One of the exciting possibilities of the results of this study would be the inclusion of continual problem discovery metrics into a user testing regime. Through continual testing of results, developers could optimize their testing to only include the required number of tests up to the desired problem discovery rate. This has the potential to concentrate testing resources on those cycles that need it rather than equally spreading resources across all cycles.

One possible scenario of use for the 4±1 rule is in time constrained agile cycles, more appropriately towards the start of projects. At this part of the project cycle, there is almost no point in discovering 100% or even 90% of possible problems if development is moving a pace which essentially wipes these problems out or replaces them with new ones.

If a project is severely budget limited, then rules of thumb such as 4±1and 10±2 will also come into play, although we would suggest that the 4±1 rule be only used on relatively simple projects.

This study was conducted in a specific project. However, the applied methodology could be applied to different project task types, different user groups and different environments to see if the rules of thumbs on the number of user tests required can be specified for different types of build projects of differing complexity and task orientation. Such analyses should use at least eight or nine user tests in their parameter estimation to eliminate the small sample size issues that presented in this study.

REFERENCES


