Benefits and Barriers of User Evaluation in Software Engineering Research

Raymond P.L. Buse
University of Virginia
buse@cs.virginia.edu

Caitlin Sadowski
University of California Santa Cruz
supertri@cs.ucsc.edu

Westley Weimer
University of Virginia
weimer@cs.virginia.edu

Abstract

In this paper, we identify trends about, benefits from, and barriers to performing user evaluations in software engineering research. From a corpus of over 3,000 papers spanning ten years, we report on various subtypes of user evaluations (e.g., coding tasks vs. questionnaires) and relate user evaluations to paper topics (e.g., debugging vs. technology transfer). We identify the external measures of impact, such as best paper awards and citation counts, that are correlated with the presence of user evaluations. We complement this with a survey of over 100 researchers from over 40 different universities and labs in which we identify a set of perceived barriers to performing user evaluations.

Categories and Subject Descriptors H.1.2 [*Information Systems Applications*]: Models and Principles—User/Machine Systems; D.0 [*Software*]: General

General Terms Experimentation, Human Factors

Keywords Human study, User evaluation

1. Introduction

Modern software engineering is an inherently human-centric activity. From requirements, architecture and design through development, testing and maintenance, the inputs, processes and outputs are primarily created, evaluated and performed by humans. While there have been many advances in automating various aspects of software engineering (e.g., specification mining, test input generation, component composition, software synthesis), major stakeholders such as developers, managers and consumers remain primarily human. It is thus perhaps surprising that of 1,718 papers surveyed from ten years of selective software conferences, only about 10% used humans to evaluate a research claim directly.

Still, user studies in software engineering are almost as old as the field itself, predating even FORTRAN. For example, in his seminal 1953–54 work on SPEEDCODING, the first higher-level language for an IBM Computer, John Backus reported on user experiences: "many problems which might require two weeks or more to program in 701 language can be programmed in Speedcoding in a few hours." [1, p.6]. Since then, such studies have not always been common. Nonetheless, we find that the use of user evaluations in research published at top venues has grown 500% since the year 2000. In the case of OOPSLA, the user evaluation rate has more than tripled since 2007. This paper explores the recent history and current state of such evaluations and identifies benefits from and barriers to performing them.

In this paper, we use *user evaluation* to mean the process of evaluating or understanding a technique, tool, or idea in terms of the needs, preferences, and abilities of humans. We collectively refer to humans involved in any stage of the software engineering process as *users*. From a research perspective, both a developer considering a new programming language feature and a consumer navigating a carefully-designed GUI are users of the technique in question.

This paper explores three main research questions:

- RQ1: *What are trends in performing user evaluations within software engineering papers?*
- RQ2: *Are there external, empirically measurable, notions of quality and impact correlated with performing user evaluations in software engineering papers?*
- RQ3: *What are the barriers software engineering researchers perceive in considering performing a user evaluation? How have they been mitigated by other researchers?*

We find, for example, that the number of papers with user evaluations is increasing, both in absolute and relative terms. With all papers considered, those with user evaluations do *not* have higher citation counts overall. However, when attention is restricted to highly-cited works, user evaluations are relevant: for example, among the top quartile of papers by citation count, papers with user evaluations are cited 40% more often than papers without. Highly-selective conferences accept a larger proportion of papers with user evaluations than do less-selective conferences. Promisingly for resource-strapped investigators, large numbers of professional developers and real-world projects are not required to

OOPSLA'11, October 22–27, 2011, Portland, Oregon, USA.

perform highly-cited user evaluations. Each of these claims is made in a statistically significant manner and is detailed in the remainder of this paper.

Overall, this paper makes several related contributions:

- We present the results of a survey of 107 researchers, highlighting the perceived difficulties and benefits of performing user evaluations. In particular, we find that 91% of participants who have performed user evaluations agreed that they gained insights from performing a user evaluation which they may not otherwise have had. We also highlight barriers to performing user evaluations identified by researchers.
- We present a classification of recent user evaluations based on 3,110 papers from ten years of selective conferences. This includes longitudinal information about how reported research has changed over time. We find that user evaluations are increasing in frequency and that papers containing human studies are associated with impact indicators (e.g., awards and greater citation counts).
- We develop a concise classification scheme for user evaluations used in software engineering research. We validate this classification scheme with both the survey of software engineering researchers and the classification of papers containing a user evaluation.
- We develop a formal model relating words in research paper text to the presence of a user evaluation described in the paper. Our model is 95% accurate, but perhaps more importantly, is a concrete example of an approach to reducing the cost of a user evaluation. Previous work in such models involved large amounts of user annotation; we train on a small amount of annotated data, and then evaluate model accuracy.

The structure of this paper is as follows. In Section 2 we present background and place our results in the context of related work. We discuss our research methodologies in Section 3. In Section 4 we present the results of a study of 3,110 recent research papers and our survey of human researchers, including identification of trends in Section 4.1, benefits in Section 4.2, and barriers in Section 4.3. We discuss the results and identify possible threats to the validity of our work in Section 5 and conclude in Section 6.

2. Related Work

This is not the first paper to analyze user studies in software engineering research. However, previous research has focused primarily on classifying experiments, whereas we focus on identifying benefits and barriers to performing user evaluations. To this end, we broaden our analysis by surveying researchers and examining previously-unused metrics such as best paper rate. Furthermore, we are not aware of any previous papers which explore empirical relationships between an evaluation strategy and proxies of impact (e.g., citation count).

Throughout this paper, we will use the terms "user evaluation", "user study", "human study", and "human evaluation" interchangeably to refer to user evaluations. User evaluation in this context must include recruitment of external participants or else the use of specific predictive human models (e.g. cognitive models). Note that user evaluation does *not* encompass studies of human-created artifacts (e.g. mining software repositories). Some reports, often labeled "case study" or "experience report", consist of investigator reports on personal experiences with self-made tools. When no external participants are involved, we do not consider the study to be a user evaluation.

Sjøeberg *et al.* identified 103 papers containing controlled experiments from a collection of 5,453 articles published in software engineering journals [24]. They discuss various facets of these experiments, such as threats to validity, whether they are replications, the subjects, and the tasks performed. They found that very few research papers contained controlled experiments, and many of those experiments may not be representative and have unaddressed threats to validity. They also found that most published replications conducted by original authors confirm the results of the initial experiment whereas most replications conducted by a new author differ with the results of the original paper, however, there were only 20 replications in the sample of 103 papers.

Lung *et al.* [17] investigated the difficulty of replicating human subjects studies in software engineering. They found many key details were omitted from the study description. They concluded that literal replication may not be the most effective strategy for validating the results of previous studies.

Tichy *et al.* [27] identified computer science papers containing empirical studies, and compared them to papers in other disciplines. They found, for example, that there were fewer such papers than expected within computer science. Shaw [21] conducted a manual review of the abstracts of paper submitted to ICSE in 2002. She concluded that field experience and realistic examples tend to be the most effective ways of validating a result.

Glass et al. [8] categorized software engineering articles based on topic, research approach, research method, reference discipline, and level of analysis. Host *et al.* [10] created a classification scheme of user evaluations based on the incentives of subjects. Their classification scheme is primarily designed to support understanding and replication. We employ a very similar classification scheme in this paper and apply it to many more studies.

Other researchers have analyzed experiments in software engineering research in the context of taxonomy validation [4]; this taxonomy was later used again to classify experiments in software engineering venues [28]. There are also a number of papers which review experiments within

particular subfields of software engineering (e.g., [5, 6, 12, 13]). Previous researchers have also drawn attention to the general lack of consideration of human factors in software engineering and programming languages research [9].

3. Methodology

In this section we describe our classification scheme for user evaluations, our large-scale empirical study of user evaluation in software engineering research, and our survey methodology.

3.1 Classification Scheme

We desired a simple, easily applicable, and semantically useful classification scheme for the types of user evaluations found in software engineering papers. We developed a novel scheme based on a review of related literature (e.g., [24]) and refined it using data from a pilot survey. Table 1 shows the classification scheme for user evaluations we adopted in this paper.

Our scheme is based on the important split between task-based studies (i.e., *Comparative*, *Observation*, *Field*) and non task-based studies (i.e., *Judgment*, *Descriptive*). In general, task-based studies are more time consuming to administer because of the additional burden of designing and observing the tasks. Not all task-based studies should be treated equally: the importance of controlled experiments (cf. [24]) motives a division between *Comparative* and *Observation* studies, and the separate explicit *Field* category was added based on pilot responses. Non task-based studies primarily involve asking participants questions about their own experiences (*Descriptive*) or participants rating artifacts (*Judgment*). This distinction is also found in previous work [18]. Finally, some research models human interaction without human participants (*Models*).

When categorizing papers, we also employ a mostly-orthogonal secondary classification scheme based on the artifacts used. The artifacts classification scheme (Table 2) is simplified from the scheme of Host *et al.* [10] and is designed primarily to capture the *motivations* of the study participants. In general, *Real* refers to an artifact that would have existed even if the user study had not taken place, *Artificial* refers to an artifact and context created for the purposes of the study, and *Isolated* refers to objects, such as code snippets, taken out of context. Of special note are user evaluations involving assignments undertaken by classroom students; we classified such projects as *Artificial*. In Section 4.1 we discuss why we elected not to include education-focused papers in our pool of software engineering papers for analysis. Note that papers can contain multiple independent studies. In some cases, studies may contain elements from multiple categories. In these cases, we choose the category that best describes the study in question.

We validated our classification scheme in two ways:

1. We successfully used it to hand-annotate 211 papers identified by our empirical analysis (Section 3.2) as containing a user evaluation.
2. We asked survey participants to identify all types of user evaluations they had administered, and provided an opportunity for specifying additional categories. None of the 64 participants who had experience performing a user evaluation specified any additional categories.[1]

Comparison with other schemes Many other researchers have proposed taxonomies of human studies and also of other empirical software engineering research methods [15]. For example, Basili *et al.* [2] present an extremely detailed classification scheme for empirical software engineering experiments.

Zelkowitz *et al.* [4] developed a taxonomy of experimentation schemes in software engineering, and validated their taxonomy by classifying software engineering papers. They identify 12 categories fitting into three broad groups: observational methods (cf. our *Observation* and *Field*), historical methods (we do not consider these to be user evaluations), and controlled methods (cf. our *Models*, *Comparative* and *Judgment*). Descriptive studies are beyond the scope of their paper.

When finalizing our classification scheme, we considered the eight strategies in four quadrants of McGrath's classic categorization of research strategies [18]. Quadrant 4 (theoretical strategies) maps to our *Models* category. The two strategies within Quadrant 3 (respondent strategies) correspond to *Descriptive* and *Judgment*. We also have an explicit *Field* category corresponding to Quadrant 1 (field strategies). The two categories from McGrath's experimental strategies Quadrant 2 map to *Descriptive* and *Observation* experiments, although McGrath divides based on the realism of the environment. This correspondence with established research strategies provides additional confidence that our scheme adequately covers the relevant possibilities.

3.2 Empirical Study

We begin by describing a manual process for characterizing the user evaluations in a research papers. We then describe how, by training on a relatively small number of human-annotated papers, we can create a descriptive textual model capable of automatically identifying, with high accuracy, all papers in the corpus containing a user evaluation. This classifier allows us to quickly hand-annotate only those papers which likely contain a user evaluation.

Using these tools we characterize the current state as well as trends over the last ten years of user evaluations in several ways. For example, we can count and measure the self-reported subject matter of such papers to gain insight into

[1] Figure 13 shows the percentage of participants who had performed a user evaluation for each of the categories.

Study Type	Description	Example
Comparative	Participants perform the same task under different situations.	*debugging code with/out a tool*
Observation	Participants are observed performing a specific task.	*case study of new language feature*
Field	Observations of people in the field performing various activities.	*counting daily interruptions*
Judgment	Participants judge one or more artifacts; don't actively solve a problem.	*heuristic evaluation*
Descriptive	Participants respond to questions, drawing on their own experiences (performing no development task).	*focus groups; experience surveys*
Models	Validation based on predictive models without human participants.	*prediction with cognitive models*

Table 1. Classification scheme for user evaluation types.

Artifact Type	Description	Example
Real	A pre-existing project is considered and the major objective of the subjects is not only to participate in the experimental study.	*in situ studies*
Artificial	Subjects consider relationships to supporting material.	*designed programming tasks*
Isolated	Objects of study are presented without additional context.	*participants consider code snippets*
No Specific	No specific artifact is used.	*most surveys etc.*

Table 2. Classification scheme for user evaluation artifacts. We adapt this scheme from Host *et al.* [10].

the sub-domains of software engineering where user studies are most applicable.

Data Set We restrict attention to five of the most prominent software engineering publication venues: The International Conference on Automated Software Engineering (ASE), The International Symposium on Foundations of Software Engineering (ESEC/FSE), The International Conference on Software Engineering (ICSE), The International Symposium on Software Testing and Analysis (ISSTA), and Object-Oriented Programming Systems, Languages and Applications (OOPSLA). In addition, and for the purpose of comparison, we also consider The Conference on Human Factors in Computing Systems (CHI) which represents a community of researchers who may also be associated with certain software engineering concerns (e.g., 18 of 107 survey respondents had submitted to both CHI and ICSE) and who care about user evaluations. For each conference we mined papers published since the year 2000 (i.e., 2000–2010). Not every conference was held every year in that window. Table 3 gives a breakdown of the 3,110 total papers we analyzed, 1,718 of which were from software engineering (i.e., non-CHI) venues.

Annotation We first manually annotated a random set of papers as training data for a classifier. For each of 100 papers, two researchers read enough of the paper to determine the presence or absence of a user evaluation (e.g., for some papers the abstract, introduction and conclusion suffice). The taxonomies introduced in Section 3.1 were employed to label each user evaluation located. The number and type of participants (students, practitioners, or both) was also recorded.

The researchers discussed the annotations for each paper until a consensus was reached. This step ensures that the classification scheme is well-defined and can be consistently applied. After completing the initial set of 100 papers together, one of the two researchers annotated an additional 82 papers to help ensure that the set was large enough for use as training data for a document classifier (i.e., to mitigate the threat of over-fitting).

Predictive Model We present a precise, empirically-derived, text-based model of research papers with user evaluations. The model determines whether a given research paper is likely to contain a user study without human intervention. We adopt this approach for two primary reasons:

First, an automatic system allows us to quickly and consistently characterize a large number of papers. Automatic classification is also highly adaptable in comparison to manual alternatives. Second, in our experience reading papers which describe user evaluations, we have often found that the abstracts alone contain no clues to indicate that a user evaluation is present. This is especially true in the case of small studies. We therefore hypothesize that those previous studies which have relied on the ability of an annotator to classify papers solely from abstracts (e.g., [21, 24]) may be subject to a systematic bias. A textual model allows us to fully inspect the complete text of each paper.

Our model is based on a term frequency vector (i.e., "bag of words") approach, where a document is characterized by a mapping from words to frequency counts. For example, a document containing the word "participant" several times may likely contain a user evaluation. To enhance the precision of this technique, we filtered out paper sections with titles containing the terms "related work" and "references" as these were found to introduce errors when, for example,

Publication Venue		Proceedings Mined (2000-2010)	Papers
International Conference on Automated Software Engineering	(ASE)	7	280
Conference on Human Factors in Computing Systems	(CHI)	11	1210
International Symposium on Foundations of Software Engineering	(ESEC/FSE)	11	331
International Conference on Software Engineering	(ICSE)	11	739
International Symposium on Software Testing and Analysis	(ISSTA)	7	192
Object-Oriented Programming Systems, Languages and Applications	(OOPSLA)	11	358
total		58	3110

Table 3. Corpus of papers and publication venues examined in this study.

Term	Predictive Power
asked	0.3519
participants	0.2259
students	0.2149
experience	0.1627
graduate	0.1549
human	0.1397
interview	0.1326
we asked	0.1326
conducted	0.1290
programming experience	0.1160

Table 4. The ten most predictive textual terms for classifying papers with user evaluations using the ReliefF method [20]. A power of 1.0 indicates that the feature is a perfect predictor, 0.0 indicates it provides no information toward the classification goal.

Classifier Evaluation		
Correctly Classified Instances	172	94.5055%
Incorrectly Classified Instances	10	5.4945%
Kappa statistic	0.8105	
Mean absolute error	0.0559	
Root mean squared error	0.2344	
Class: No User Evaluation		
Precision	0.9603	
Recall	0.9732	
F-measure	0.9667	
Class: Yes User Evaluation		
Precision	0.8814	
Recall	0.9123	
F-measure	0.8966	

Table 5. Evaluation of the decision tree classifier used to determine if a paper contains a user evaluation. The model has very high accuracy (94.5%) and behaves similarly on yes-instances and no-instances.

authors discussed previous studies they they did not conduct. To learn which words are predictive we use a feature selection technique called ReliefF [20], which characterizes the predictive power of features without assuming conditional independence. Table 4 enumerates the ten most predictive terms. As potential features we consider both individual terms as well as two term tuples (e.g., in addition to "asked" we use phrases such as "we asked"). For our model, we selected all terms that have an estimated predictive power greater than 0.01 (approximately 30 terms). A principle components analysis indicates that to explain 90% of variance, 44 principle components are necessary.

To test our model and check for over-fitting, we used leave-one-out cross validation on the 182 papers in our training corpus. We experimented with several classifiers, including Logistic and Bayesian models, but found C4.5 decision trees [19] to give the best performance: a correct classification rate of 94.5% with an error rate of 5.5%. Table 5 presents additional statistics relevant to the performance of the model.

We employed the trained model to label each of the remaining 1,718 software engineering papers in our database, identifying 294 which are likely to contain a user evaluation. The two researchers who annotated the original set of 182 papers then annotated those 294 papers, applying the same annotation rubric to reveal the type and size of each study. In all, 211 user evaluations were positively identified (see Table 6). In the following discussion, we leave out the *Models* category since we did not identify any papers which contain a model-based user evaluation. Nonetheless, almost 20% of survey participants who had performed a user evaluation before said they had previously performed a model-based evaluation (Figure 13). Although model-based evaluations appear in human computer interaction papers (e.g. [26]), they are less commonly reported in recent software engineering research literature.

False positives are rejected manually and are thus not a concern. False negatives, which we characterize by inspecting our training data, can be described as minor. Such studies have little descriptive text and so are difficult to detect simply by means of word counts. For an indicative example, in a paper by Egyed published in ICSE 2006 [7] the following sentence appears: "Yet, in interviews with the engineers we were told that at no time they felt delays of any kind." While

Paper Category	# of Papers ($n = 3110$)
SE Papers	1718
Study Candidate SE Papers	1100
SE Papers w/ User Evaluation	211
SE Education Papers	50
Study/Artifact Category	**# of Studies ($n = 211$)**
Comparative	48
Observation	51
Field	23
Judgment	25
Descriptive	64
Real	62
Artificial	74
Isolated	44
No Specfic	31

Table 6. A breakdown of major paper categories considered throughout this study. "Study Candidate" papers are those which involve a topic wherein at least 10% of paper contain a user evaluation (see Section 4.1).

this qualifies as a user evaluation by our definition, insufficient descriptive text is present for our automated model to correctly classify it. Thus, while our estimated false negative rate is approximately 9%, the lack of descriptive text indicates that in very few of these cases is the user evaluation tied to a major claim of the paper. We conclude that the vast majority of user evaluations corresponding to a primary paper claim are correctly identified by our approach.

3.3 Survey

To better understand the perceptions of software engineering researchers related to user evaluation, we conducted a survey of authors of papers at top-tier software engineering venues in the past 10 years (see list of venues in Table 3).

Employing the classifier described in Section 3.2, we identified researchers who had and had not performed a user evaluation during the 10-year window. We used a stratified random sampled of 450 candidate participants (200 who had performed one and 250 who had not). Each researcher was sent an email inviting them to participate in the anonymous web-based survey. We also sent two follow-up emails to candidates who did not respond to the survey when it was initially sent out. As a repayment for participation, we entered participants in a drawing for a $25 USD gift certificate. We have awarded this certificate to one of the participants.

107 researchers completed the survey (24% response rate). Those who our classifier determined had previously conducted a user evaluation responded at a higher rate than others (overall, 61.7% of participants had performed a user evaluation). Figure 1 shows the number of user evaluations performed by survey participants.

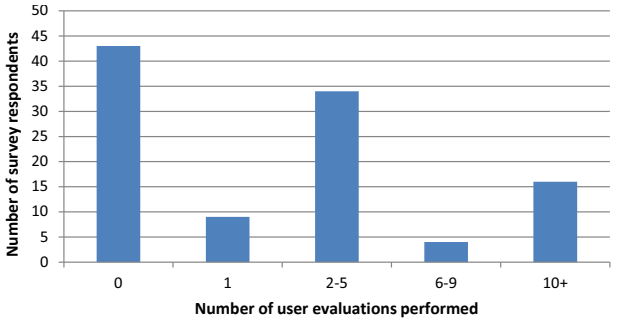

Figure 1. Number of user evaluations performed by 107 survey respondents.

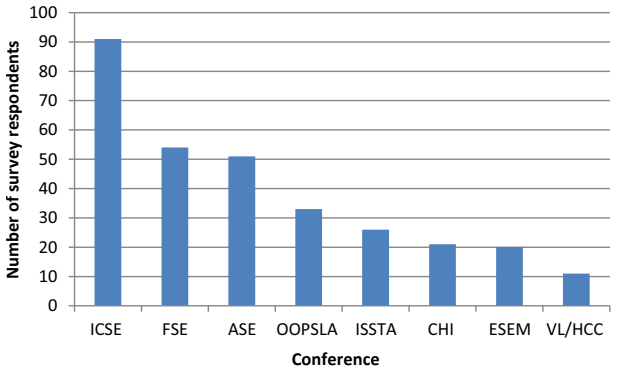

Figure 2. For each given SE-related venue (x-axis), the percentage of 107 survey participants who had submitted at least one paper to that venue within the last ten years.

Of the participants, 54% were academic faculty and 24% were students, while another 22% were non-academic (e.g., from research labs). Participants responded from more than 40 different institutions. 30% of participants reviewed more than 10 submissions to major research conferences or journals last year. Figure 2 shows the breakdown of what percentage of participants submitted to different SE-related conferences in the past 10 years.[2] Most participants (88%) had submitted a paper to ICSE.

We asked participants to rate a variety of statements about user evaluation such as "the benefits of user evaluation generally outweigh the costs" on a five-point Likert scale ranging from Strongly Agree to Strongly Disagree. We then asked participants to identify barriers to using user evaluation in their own research. We also asked participants whether they had performed a user evaluation in the past: participants who had were asked additional questions about the last user evaluation they had performed.

When reporting results from this survey, we use a Chi-Square test for independence to compare the Likert-scaled perceptions of user evaluation against whether participants

[2] The International Symposium on Empirical Software Engineering and Measurement (ESEM) And IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC) were mentioned by survey participants in free-form response but are not studied in this paper.

had ever administered a user evaluation. This test is used to identify statistical association between two categorical variables. In situations where the number of participants in one category was too small, we instead used the Fisher-Irwin test to check for statistical significance [3], which can be extended beyond 2 x 2 matrices. We deem a result with a p-value below 0.05 to be statistically significant.

We piloted the survey with five graduate students and three researchers to verify that the questions were clear; pilot participants are not counted among the 107 and their results are not included. This iteration helped us identify a list of potential barriers to performing user evaluations, and refine the classification scheme for different types of user evaluations.

4. Results

In this section we analyze our annotated corpus of research papers as well as responses to our survey. We begin by exploring trends in user evaluations by subject matter and over time (Section 4.1). Finding that the use of such evaluations is rapidly increasing, we quantify and qualify some of the important benefits of user evaluations (Section 4.2). Finally, we investigate the most common perceived barriers to such research (Section 4.3).

4.1 Trends

We begin by exploring recent trends in user evaluations. First, we investigate which research topics user evaluations are commonly associated with. Then we break down trends in such evaluations over time.

Topic Trends To measure the impact of user evaluations, we must first characterize their domains — the research subfields in which they are employed. This allows us to control for those topics for which a user evaluation is an appropriate strategy. For example, if papers on the topic of "formal verification", or on other topics which generally do not contain user-studies, are highly cited (or not highly cited), this could prevent us from detecting significant trends and evaluating the hypothesis (e.g., user evaluations are correlated with higher citation counts).

To identify topics which pertain to user evaluation we use the ACM Classification system: generic categories are typically added manually, by paper authors, when papers are submitted for publication. Figure 3 shows the top 30 most commonly used descriptors from this system. Each is labeled with the percentage of papers in our data set bearing that description that contain a user evaluation. As expected, papers pertaining to theory and verification contain fewer user evaluations, while subjects such as programming environments contain many. Note that we do not wish to suggest which topics *should* contain user evaluations, we only present this characterization for the purpose of controlled analysis. Nonetheless, there are many popular topics containing few user studies which we believe would likely benefit from them. In particular, we were surprised at the low percentage of papers under "d.2.5 testing and debugging" that contained user evaluations.

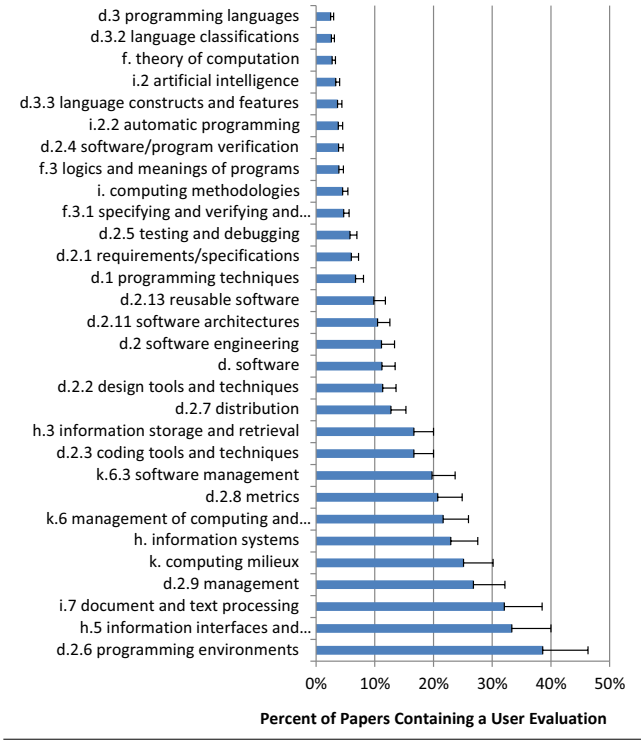

Figure 3. Percent of papers containing a user evaluation with estimated false negative rate, in each of the top 30 most commonly used ACM category descriptors.

We have identified 19 subject and sub-subject areas for which at least 10% of the associated papers contain human studies. We term these *study candidate* subjects and use this set in subsequent experiments. In all, 1,100 papers, or about 57% of the software engineering papers we studied, are annotated with one of these descriptors and are thus a member of the study candidate category.

Finally, we also identify those papers in the area of software engineering education research. Because human studies in this area are very different from standard user evaluations (e.g., research on enrolled students about the effectiveness of a particular teaching technique) we conservatively elect not to consider them in our pool of software engineering papers. We match the title, abstract, and subject descriptors against the term "education" in order to identify such papers. A total of 50 were located in our dataset.

Longitudinal Trends We employ the classification methodology described in Section 3.2 to explore trends in rates of user evaluations over time. We characterize by *rate* because the total number of publications in software engineering at the conferences in our dataset has increased greatly over the last decade (nearly doubling from 127 papers published in 2000 to 232 in 2010).

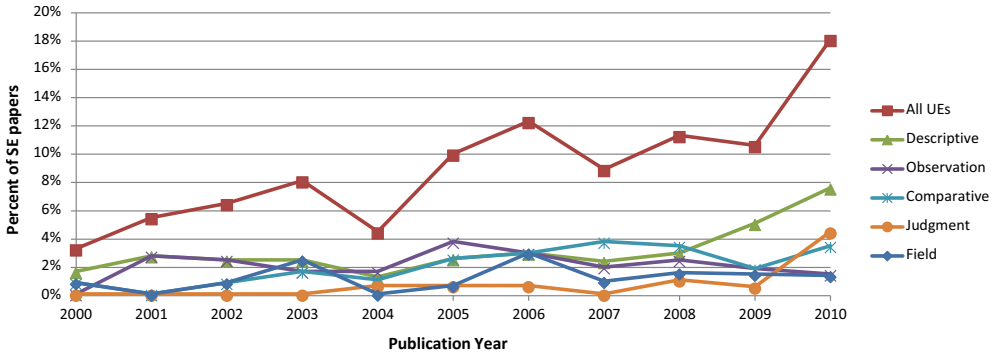

Figure 4. User evaluation subtype trends over time. Note the increase evaluations, especially since 2004.

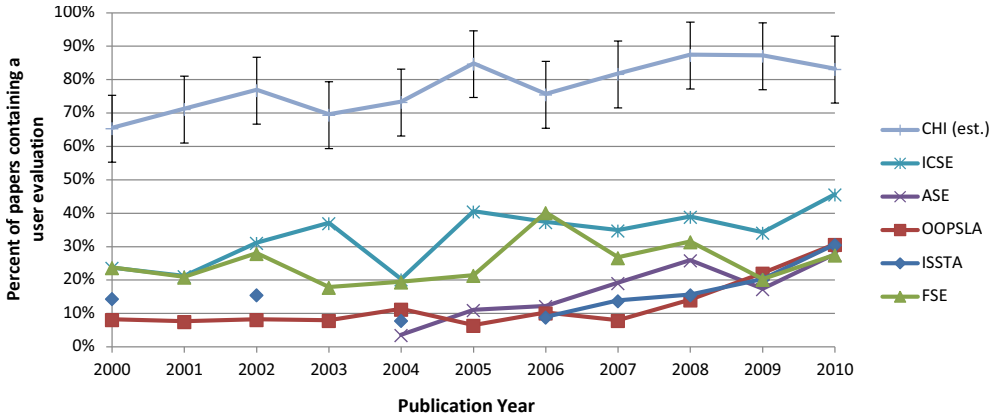

Figure 5. User evaluation trends over time by conference. Note the steady increase in such studies at many software engineering conferences including ASE, ISSTA, and OOPSLA.

We explore whether user evaluations are employed more often today than ten years ago:

H1: *The number of papers with user evaluations is increasing, in both absolute and relative terms.* **Supported.**

Figure 4 indicates a significant positive trend in the number of user evaluations in SE research. Where such studies comprised as few as 3.5% of papers in 2000, about 18% of papers published in 2010 contain a user evaluation (an increase of over 500%). The number of *study candidate* papers (i.e., papers in subject areas with more than average human studies) have not increased significantly as a fraction of all papers during that interval. While *Descriptive*, *Comparative*, and *Judgment* type studies are preformed more frequently today, we find that *Observation* and *Field* type studies have not shown significant growth. In general, this would indicate a trend toward controlled and empirical studies over studies which are primarily qualitative and often require direct access to professional developers.

As indicated by Figure 5 this trend is particularly strong in the case of ASE, ISSTA, and OOPSLA which each show large increases in user evaluations in recent years. ICSE show a lesser but significant positive trend and FSE exhibits noisy but relatively constant level of user evaluations for the study period.

4.2 Benefits

Having observed that use of user evaluations is on the rise, we now explore the question of *why* this is the case. In particular, we seek to qualify and quantify some of the potential benefits of user evaluations in software engineering research by exploring correlations with common influence metrics. We test whether the presence of such studies show a significant correlation with citation count (as reported by ACM Digital Library), best paper awards, and selectivity of conference. While none of these metrics are precise indications of quality, they are used throughout the research community as proxies when evaluating the impact of work and are thus important even in isolation.

Benefit: Impact

H2: *Papers with user evaluations have higher citation counts.* **Rejected.**

We first investigate citation rates between papers with and without a user evaluation (as predicted by our automated model). Figure 6 shows that, since 2002, papers with such studies have usually been cited slightly more frequently

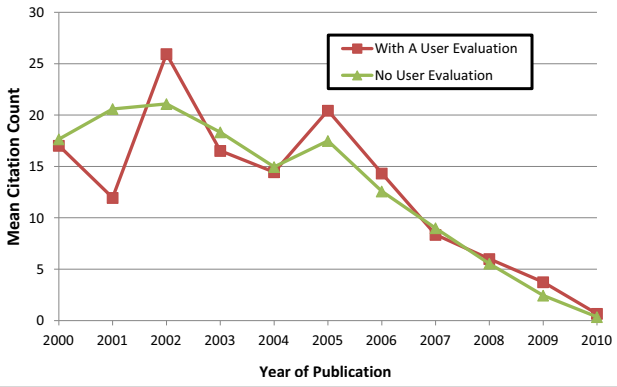

Figure 6. No significant difference on aggregate between the citation rate of papers with and without user evaluations.

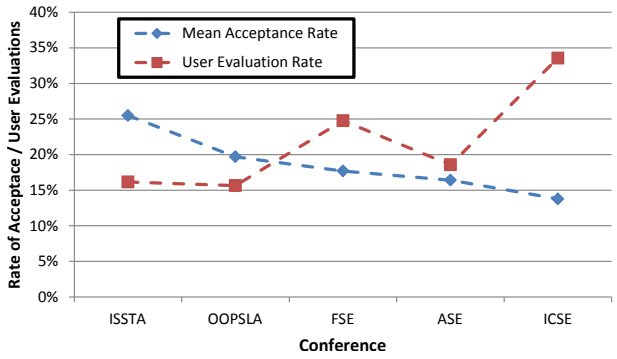

Figure 7. More selective conferences tend to publish a larger proportion of papers containing user evaluations.

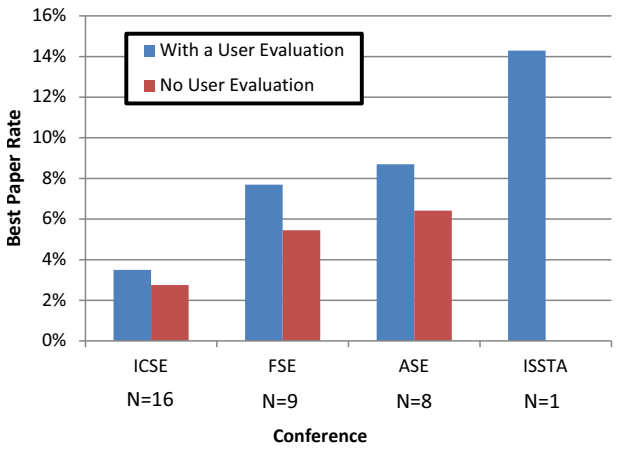

Figure 8. *Study candidate* papers (in categories where at least 10% of papers contain human studies) containing user evaluations occur more commonly as ACM Distinguished Papers.

than papers without such studies. However, this difference is small and not statistically significant.

H3: *Among higher-impact papers, papers with user evaluations have higher citation counts.* **Supported.**

In Figure 9 we tease apart user evaluations, both by type of evaluation, and also by citation percentile (i.e., where the paper ranks among other papers by citation count). We measure citations per year so as not to bias our study toward older papers. We restrict attention to study candidate papers (without this restriction, the trend we observe here is less significant). Also, we do not include papers written in 2010, since they have not been published long enough to be cited significantly. Figure 9 shows that while papers that are not cited often (i.e., the 25th percentile) show no significant difference with respect to whether they contain a user evaluation, higher-impact papers (i.e., 50th and 75th percentile papers) show an appreciable and significant difference. For example, among high-impact papers (75th percentile), papers containing a user evaluation are cited 3.9 times per year, while papers without (the "No UE" bullet) are cited an average of 2.8 times per year. This difference is statistically significant with t-test $p < 0.01$. To put it another way, while papers without a user evaluation were ranked as the third most widely cited amongst 25th percentile papers, they are ranked as least cited amongst 75th percentile paper.

We conclude then that the presence of a user evaluation, in and of itself, is not sufficient to correlate with a paper's impact. However, widely cited papers are often viewed as having significant impact, and among such papers the presence of a user study does correlate with citation count in a significant manner. We also note that light-weight *Judgment* and *Descriptive* studies, which are generally less expensive to conduct than many other types of studies, consistently rate high by this metric.

We now consider the relationship between of user evaluations and award-winning research. We mined the history of ACM Distinguished Paper awards, finding 75 papers in our database associated with such an award. Of those, 11 contain user evaluations. After controlling for *study candidate* papers (see Section 4.1) we find that those papers containing a user evaluation are about 30% more likely, on average, to win a best paper award than other papers (Figure 8).

Benefit: Selectivity of Publication Venue

H4: *Highly selective conferences tend to publish a larger proportion of papers containing user evaluations.* **Supported.**

Finally, we consider conference acceptance rate as third proxy for impact. Figure 7 shows that mean acceptance rate is negatively correlated with the rate of user evaluations, indicating that the most selective conferences publish a higher proportion of papers containing these studies.

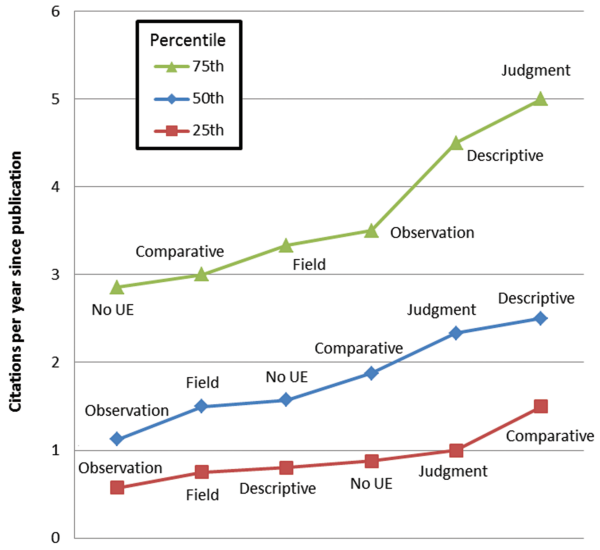

Figure 9. The number of citations per year since publication for 5 types of user evaluations as well as papers without studies divided into 3 percentiles. Note that for paper in the 25th percentile of citations (papers which are cited relatively infrequently) the presence of user evaluations has little impact on citation rate. However, for 75th percentile papers (those which are cited relatively often), papers with user evaluations are cited much more frequently than those without such studies.

Benefit: Insights 91% of survey participants who had performed a user evaluation agreed that they gained insights from performing a user evaluation which they may not otherwise have had. We found that participants who have performed a user evaluation are more likely to agree that the benefits of user evaluation generally outweigh the costs: 80% of participants who had performed a user evaluation agreed, compared to 50% of those who have not. Most participants, even those who had not conducted a user study, agreed that a user evaluation increases the impact of a paper (82%), and that an appropriately designed user evaluation contributes strongly to the likelihood of publication in a major conference (76%). However, participants who have performed a user evaluation before were more likely to strongly agree to these statements. The above differences are all statistically significant, with $p < .003$ in all cases.

4.3 Barriers

We have observed that user evaluations are correlated with a variety of important influence metrics. Nonetheless, historically only about 10% of software engineering research papers contain such studies. To better understand this disconnect we now investigate the barriers to user evaluations. We employ survey data to characterize perceived barriers and then use data about actual papers to hypothesize ways that researchers can mitigate some of these barriers.

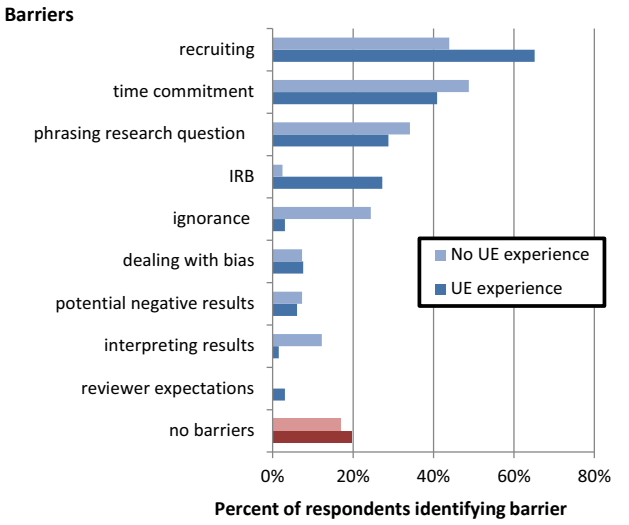

Figure 10. Barriers identified by participants who have or have not performed a user evaluation.

Most survey participants (84%) agreed that "user evaluation is difficult." However, there was not a statistically significant difference between whether participants had performed a user evaluation and their response to this question. Instead of a general perception of difficulty, there are specific barriers which prevent software engineering researchers from engaging in user evaluations.

The first step in overcoming barriers is identifying them. Through the survey we identified five major and four minor potential barriers for software engineering researchers in performing user evaluations (Figure 10). About 20% of participants cited that there was no barrier to them performing a user evaluation; this percentage is similar between participants who had or had not performed a user evaluation in the past. In the following subsections, we will describe the most prevalent barriers and discuss some mitigating factors. We note that almost 25% of participants who had never performed a user evaluation identified the fact that they did not know how to perform such an evaluation as a barrier to using user evaluation in their research. This highlights the need for more resources related to performing user evaluations in software engineering research.

Perceived Barrier: Recruiting Our survey respondents indicated that they perceive that user evaluations are difficult to conduct for a variety of reasons, but especially because of concerns related to recruiting. Almost 60% of participants cited recruiting as a barrier to employing user evaluation in their own research. Unfortunately, we noticed that very few of the user evaluation papers we read and annotated discuss how the authors handled participant recruitment.

Perceived Barrier: Experimental Design and Time The second two most prevalent barriers were feeling that performing a user evaluation takes too long (44% of participants), and not being sure how phrase the research question

(31%) in order to design a user evaluation. It is very difficult to measure how long studies took to administer from reading the high-level study description in a research paper; this information is typically not included. However, we believe that our classification scheme does provide some indication of time commitment in that task-based evaluations (*Observation*, *Comparative*, *Field*) are often more time consuming than non task-based evaluations (*Descriptive* and *Judgment*).

Perceived Barrier: Institutional Review Board (IRB) An *institutional review board* (or *independent ethics committee* or *ethical review board*) is a committee that reviews and oversees studies involving human subjects to ensure they are ethical and regulatory. In the United States, they are related to the National Research Act of 1974. Only one participant who had not performed a user evaluation identified the institutional review board (IRB) as a barrier, compared with 18 participants who had previous user evaluation experience. We believe that many researchers without experience using human subjects do not know about the role of the IRB in such research. Unfortunately, software engineering researchers rarely mention the IRB when describing studies, or discuss any challenges in getting IRB approval. We located 14 papers in the corpus that mentioned the IRB, but 12 out of those 14 were published at CHI.

Other Perceived Barriers Our survey also included a free-form text field for describing barriers. Three major additional types of barriers were mentioned frequently:

1. Reviewer expectations

   "Reviewers who do not do this type of research, yet have unrealistic expectations on those who do (as evidenced by their review comments)."

   — Survey Participant

2. Dealing with biases

   "User evaluations are difficult, and can be subject to bias and/or small numbers. As such, they are not always appreciated ..."

   — Survey Participant

3. Interpreting the results

   "It seems difficult to do in a meaningful way."

   — Survey Participant

Mitigating Barriers The two largest barriers identified were recruitment and the time commitment. To address these barriers, we investigate the type of user evaluations that participants employed in software engineering research papers and identify studies employing lightweight methodologies.

In this section we investigate the populations typically recruited for user evaluations and characterize them by students vs practitioners and by number of participants. We conjecture that, in general, smaller studies are easier to recruit for, as are studies of students who are typically accessible for academic researchers. We also conjecture that

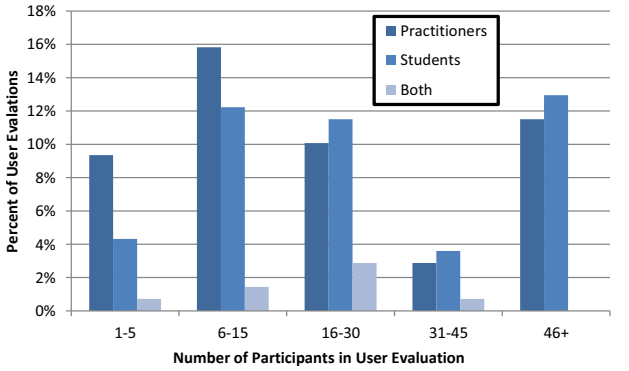

Figure 11. A histogram which breaks down the studies found in our database by number and type of participant. We find, for example that many studies are conducted utilizing fewer than 15 participants.

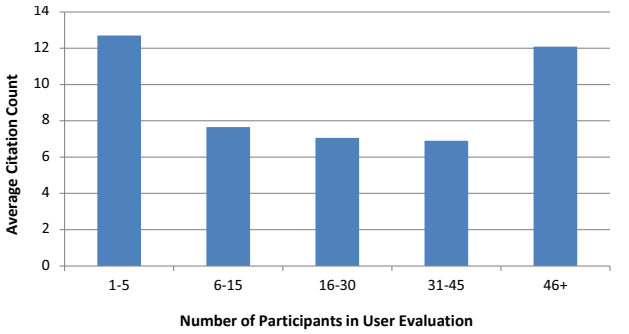

Figure 12. A histogram which relates average number of citations to the number of participants in a study. Except for very large studies (over 45 participants), there is a significant downward trend (Pearson's r p-valve < 0.05) in citation count. This suggests that small studies are often sufficient to achieve impact goals.

smaller studies take less time to administer. Of course, many other factors can substantially influence the time commitment required for a user evaluation. To explore this notion more deeply, we also investigate the roll of study types and artifact types. In particular, we conjecture that some types of studies (e.g., controlled studies involving real artifacts) can be more time consuming to conduct than some other types (e.g., judgment studies with artificial or even isolated artifacts).

H5: *Large numbers of professional developers are needed for an effective user evaluation.* **Rejected.**

In Figure 11 we aggregate data on user evaluations. We find that studies are conducted with a variety of ranges and with students and practitioners at about equal frequency. Although students may not always be appropriate subjects [22], they may be suitable subjects for many areas [25]. Notably, a majority of studies involved fewer than 20 participants and studies with less than ten are not uncommon. This finding is validated by our survey data in which the median number of

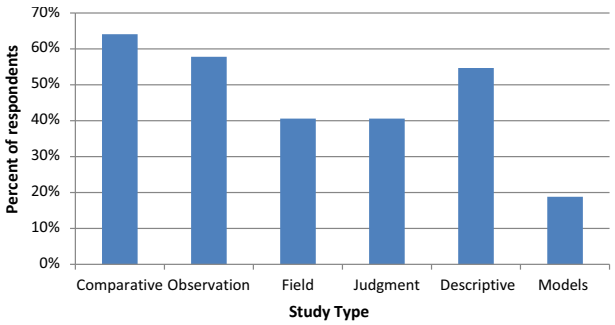

Figure 13. Percent of participants who had *ever* performed a user evaluation of the listed category.

subjects reported for the last user evaluation performed was about 20.

> *"Even a modest user evaluation (e.g. watch three friends use your software) is tremendously insightful. Just do it!"*
>
> — Survey Participant

Since studies with few participants are common, the question then becomes whether these small studies are as impactful as larger ones. Figure 12 shows that, excluding the largest studies (those with more than 45 participants) average citation count actually exhibits a negative correlation with participant count; small controlled studies may be as valuable as larger studies (at least from a citation correlation standpoint).

> H6: *Heavyweight techniques are needed for an effective user evaluation.* **Rejected.**

Study type can play a significant roll in the difficulty of experimental design and time commitment. For example, if one wishes to evaluate the utility of an engineering tool one might choose between several kinds of studies. Heavyweight options include:

- *Comparative* — Participants use the tool to preform some representative task in a realistic setting. Their performance is compared to a control group who are not given the tool.
- *Observation* — Participants use the tool to preform some representative task in a realistic setting. Their performance is observed.
- *Field* — Real developers adopt the tool into their workflow.

Lightweight options include, for example:

- *Judgment* — Participants are shown sample output of the tool and asked whenever they think it would be useful.
- *Descriptive* — Participants are asked whether a tool matching the description of the tool in question would likely be useful to them.

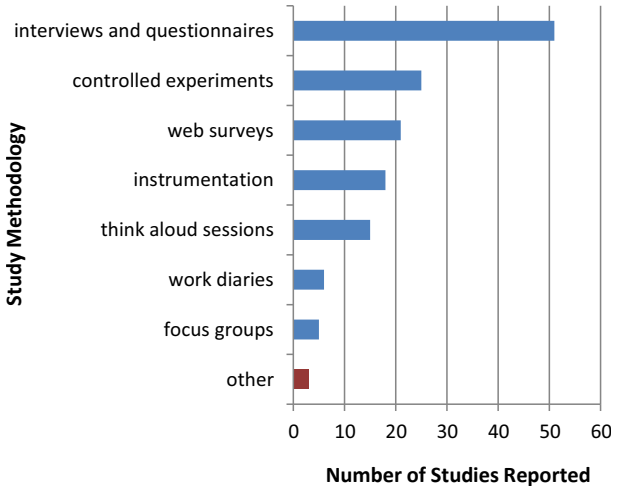

Figure 14. Methodologies employed in the last user evaluation.

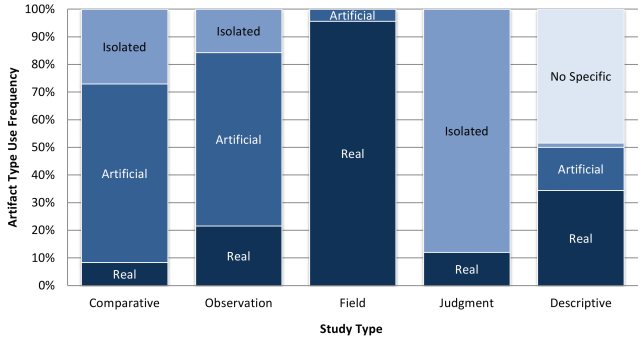

Figure 15. Shows the percentage of study types associated with each of four artifact types. For example, most judgment-based studies utilized isolated artifacts, some utilized real ones, but none utilized either artificial or no specific artifact.

Figure 13 shows the percentage of survey participants who had performed a user evaluation for each of these categories. We also asked participants to list the methodologies employed in their last user evaluation; the results are shown in Figure 14. The results support the hypothesis that heavyweight studies are not always required: for example, interviews, questionnaires, and web surveys are commonly employed. From this data we surmise that there exists many popular means of conducting user studies which do not necessarily require a large time commitment.

> H7: *Heavyweight user evaluations have higher citation counts on average.* **Rejected.**

We find that the more lightweight study categories (*Judgment* and *Descriptive*) are actually often correlated with higher average citation counts (Figure 9). We also explore artifacts associated with studies. Figure 15 shows the percentage of each type of study conducted with real, artificial, isolated, and no specific artifact. Complete descriptions of artifact types can be found in Section 3.1. We find that user

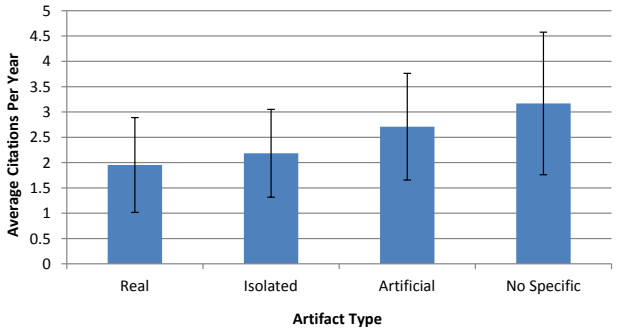

Figure 16. Average number of citations per year since publication for papers containing studies incorporating each of four artifact types. Error bars indicate standard deviation. A two-tailed T-test suggests that there is no significant difference between the mean citation rate of studies utilizing *real* artifacts and *isolated* ones. There is a significant difference between *real* and *artificial* as well as *real* and *no specific*.

evaluations are not limited to real projects. On the contrary, researchers often use artificial or isolated artifacts when conducting studies.

As compared with real artifacts, these can often allow researchers to more easily translate research questions into successful experiments. For example, rather than a study in which participants must become familiar with a real (and often complex) system to preform some task of interest, one might instead elect to conduct an experiment where the task artifact is artificially simplified. While designing an artificial project may take time upfront, it can have several important advantages. For example, it may reduce training time, simplify recruiting, and, perhaps more importantly, it can allow the researcher greater control over confounding factors. Moreover, Figure 16 indicates that these artifacts actually are correlated with greater numbers of citations, indicating that impact may not need to be sacrificed for this type of experimental control.

Finally, for more information on performing user evaluations, the reader is referred to a variety of books [16], articles [14, 15, 18, 23], and cards [11] to help in deciding between different evaluation methodologies.

5. Threats to Validity

Internal validity. In this study we identified and explored a number of correlations between user evaluations and external metrics of quality and impact. It is important to note, however, that we do not know the extent to which such correlations indicate causal relationships. Indeed, there are many factors which contribute to, say, the citation count of a paper. We cannot say that simply including a user evaluation is sufficient to yield a greater citation count. Nonetheless, qualitative feedback from our survey would tend to support such a claim.

External Validity. We have endeavored to mitigate threats to external validity by studying a large sample of papers from selective and respected publication venues. We mitigate potential bias in our annotation methodology by borrowing from previous rubrics where possible and by using two annotators to develop a consistent classification scheme.

Intentional Validity. In this paper we used a number of proxy metrics for paper impact and quality, including selectivity of publication venue, citation count, and awards. It is important to note that none of these metrics measure paper impact or quality directly. Groundbreaking research, for example, may not be cited often simply because few active researchers are working on related topics. The metrics were selected primarily because they are objective and are known to carry considerable weight, for better or for worse, in the research community. In other words, these metrics are valuable independent of whether they are truly reliable estimates of quality research (an important question that is beyond the scope of this paper).

6. Conclusion

User evaluations remain a critical pillar of software engineering research, but their associated benefits and barriers are imperfectly understood. Using a corpus of over 3,000 papers spanning ten years, we studied subtypes of user evaluations (e.g., coding tasks vs. questionnaires), and related user evaluations to paper topics (e.g., debugging vs. technology transfer). We identified the 19 "study candidate" subject areas most likely to contain user evaluations and found that the number of papers with user evaluations is increasing, both in absolute and relative terms. We also performed a survey of over 100 researchers from over 40 different universities and labs, identifying perceived barriers to performing user evaluations.

With all papers considered, those with user evaluations do *not* have higher citation counts overall. However, when attention is restricted to highly-cited works, user evaluations matter: for example, among the top quartile of papers by citation count, papers with user evaluations are cited 40% more often than papers without. Highly-selective conferences accept a larger proportion of papers with user evaluations than do less-selective conferences. We identified nine concrete barriers researchers identify to performing user evaluations, with recruitment as the most common barrier. Promisingly for resource-strapped investigators, large numbers of professional developers and real-world projects are not required to perform highly-cited user evaluations.

Acknowledgments

The authors are indebted to Andrew Begel for insightful conversations and substantial commentary on an early draft of this work. We thank Craig Anslow for suggested improvements to a later draft. We would also like to thank all of our survey responders for participating. The authors gratefully acknowledge the support

of NSF grants CCF 0954024 and CCF 0905373, AFOSR MURI grant FA9550-07-1-0532, AFOSR grant FA8750-11-2-0039, and DARPA grant FA8750-11-2-0039.

References

[1] J. W. Backus. The IBM 701 Speedcoding system. *Journal of the ACM*, 1:4–6, January 1954.

[2] V. Basili, R. Selby Jr, and D. Hutchens. Experimentation in software engineering. In P. Oman and S. L. Pfleeger, editors, *Applying Software Metrics*. 1997.

[3] I. Campbell. Chi-squared and Fisher-Irwin tests of two-by-two tables with small sample recommendations. *Statistics in Medicine*, 26:3661–3675, 2007.

[4] T. Couronne, M. Zelkowitz, and D. Wallace. Experimental validation in software engineering. *Information and Software Technology*, 39(11):735–743, 1997.

[5] I. Deligiannis, M. Shepperd, S. Webster, and M. Roumeliotis. A review of experimental investigations into object-oriented technology. *Empirical Software Engineering*, 7(3):193–231, 2002.

[6] T. Dybå and T. Dingsøyr. Empirical studies of agile software development: A systematic review. *Information and Software Technology*, 50(9-10):833–859, 2008.

[7] A. Egyed. Instant consistency checking for the UML. In *International Conference on Software Engineering*, 2006.

[8] R. Glass, I. Vessey, and V. Ramesh. Research in software engineering: an analysis of the literature. *Information and Software Technology*, 44(8):491–506, 2002.

[9] S. Hanenberg. Faith, hope, and love: an essay on software science's neglect of human factors. In *Object-Oriented Programming, Systems, Languages, and Applications*, 2010.

[10] M. Host, C. Wohlin, and T. Thelin. Experimental context classification: incentives and experience of subjects. In *International Conference on Software Engineering*, 2005.

[11] IDEO. *Method Cards: 51 Ways to Inspire Design*. William Stout, 2003.

[12] M. Jørgensen. A review of studies on expert estimation of software development effort. *Journal of Systems and Software*, 70(1-2):37–60, 2004.

[13] N. Juristo, A. Moreno, and S. Vegas. Reviewing 25 years of testing technique experiments. *Empirical Software Engineering*, 9(1):7–44, 2004.

[14] B. Kitchenham. Evaluating software engineering methods and tools. Part 1: the evaluation context and evaluation methods. *ACM SIGSOFT Software Engineering Notes*, 21(1):11–14, 1996.

[15] H. Lam, E. Bertini, P. Isenberg, C. Plaisant, and S. Carpendale. Seven guiding scenarios for information visualization evaluation. Technical Report 2011-992-04, University of Calgary, 2011.

[16] J. Lazar, J. H. Feng, and H. Hochheiser. *Research Methods in Human Computer Interaction*. Wiley, 2010.

[17] J. Lung, J. Aranda, S. M. Easterbrook, and G. V. Wilson. On the difficulty of replicating human subjects studies in software engineering. In *International Conference on Software Engineering*, 2008.

[18] J. McGrath. Methodology matters: Doing research in the behavioral and social sciences. In *Human-computer interaction*, pages 152–169. Morgan Kaufmann Publishers Inc., 1995.

[19] J. R. Quinlan. *C4.5: programs for machine learning*. Morgan Kaufmann Publishers Inc., 1993.

[20] M. Robnik-Šikonja and I. Kononenko. Theoretical and empirical analysis of ReliefF and RReliefF. *Machine Learning*, 53:23–69, October 2003.

[21] M. Shaw. Writing good software engineering research papers: minitutorial. In *International Conference on Software Engineering*, 2003.

[22] D. Sjoberg, B. Anda, E. Arisholm, T. Dyba, M. Jorgensen, A. Karahasanovic, E. Koren, and M. Vokác. Conducting realistic experiments in software engineering. In *Empirical Software Engineering and Measurement*, 2003.

[23] D. Sjoberg, T. Dyba, and M. Jorgensen. The future of empirical methods in software engineering research. In *Future of Software Engineering Workshop (FoSER)*, 2007.

[24] D. Sjøeberg, J. Hannay, O. Hansen, V. Kampenes, A. Karahasanovic, N.-K. Liborg, and A. Rekdal. A survey of controlled experiments in software engineering. *IEEE Transactions on Software Engineering*, 31(9):733 – 753, 2005.

[25] M. Svahnberg, A. Aurum, and C. Wohlin. Using students as subjects-an empirical evaluation. In *Empirical Software Engineering and Measurement*, 2008.

[26] L. Teo and B. John. Cogtool-Explorer: towards a tool for predicting user interaction. In *Conference on Human factors In computing systems (CHI)*, 2008.

[27] W. Tichy, P. Lukowicz, L. Prechelt, and E. Heinz. Experimental evaluation in computer science: A quantitative study. *Journal of Systems and Software*, 28(1):9–18, 1995.

[28] M. Zelkowitz. Techniques for Empirical validation. In V. Basili, D. Rombach, K. Schneider, B. Kitchenham, D. Pfahl, and R. Selby, editors, *Empirical Software Engineering Issues. Critical Assessment and Future Directions*, pages 4–9. Springer, 2007.