2016 Asia-Pacific Conference on Software Engineering, Hamilton, New Zealand, Dec 6-9 2016, (c) IEEE 2016

What Influences Usability Defect Reporting? – A Survey of Software Development Practitioners

Nor Shahida Mohamad Yusop, Jean-Guy Schneider
School of Software and Electrical Engineering
Swinburne University of Technology
Melbourne, Australia
{nmohamadyusop, jschneider}@swin.edu.au

John Grundy, Rajesh Vasa
Faculty of Science, Engineering and Built Environment
Deakin University
Melbourne, Australia
{j.grundy, rajesh.vasa}@deakin.edu.au

***Abstract*— Software development organizations invest in test automation tools and methods to optimize defect discovery rates. The true value of these tools is realized when the defects are addressed before release, and hence good quality defect reports are critical. We describe a survey we conducted to better understand usability defect reporting, in particular, influences on the quality of usability defect reports. We analyze feedback from nearly 150 software developers and usability defect reporters and identify key determinants of quality defect reports, aspects of usability defects that are challenging to report and directions for future research into usability defect reporting tools to improve usability defect reports quality.**

Keywords*— *usability defect reporting; role of the reporter; experience; knowledge; automation tools; defect discovery methods;

I. Introduction

According to prior studies [1], [2], "good" and valuable defect reports should be technically complete – contain no gap between expected and actual results, provide accurate steps to reproduce defects, and be supported by specific supplementary evidence. Studies have shown that the quality of defect reports is one of the most important attributes of a good tester and defect reporter [3]. Often, the reporters' ability to clearly communicate the defect determines the probability of resolution. Previous studies report that the overall quality of the defect reports are far from satisfactory. Common issues include incomplete information [1], [2], lack of clarification and focus [4], [5], mixed data [6] and mismatch information [1] in defect descriptions. A poorly-written defect report negatively affects communication between reporters and software developers.

The majority of software defect reporting research has been devoted to the enhancement and improvement of new techniques and tools for software defects in general [6], [7], [8], [9]. While it is recognized that usability defects are different to many other kind of software defects [10], [11], little has been done to understand effective ways to describe the subjective nature of usability defects. Our earlier work on usability defects have found that existing open source defect repositories are inconvenient to support usability defects – too complex for end users, limited types of attachment files, lack of integrated tools to capture usability information, and limited usability keywords and options to classify usability defects [12].

There are several factors that influence and impact on the quality of a defect report – with the key areas being skilled people supported by appropriate tools and methods. For example, [13], [14], [15] explicitly investigate the importance of tester's knowledge in performing exploratory software testing. Kettunen at al. [16] also report domain knowledge as the most emphasized area of testers' expertise and point out that the role of technical knowledge is particularly important in the agile development context. In a survey on the effect of experience, Merkel and Kanij [3] identified expertise in the problem domain and knowledge of specific testing techniques were a significant factor influencing the performance of software testers. Beer et al. [17] and Poon et al. [18] reported the impact of the testers' experience on improving testing strategies - test case design, regression testing, and test automation. In the context of usability studies, F∅lstad [19] reported that usability defects found by work-domain experts were classified as more severe and received higher priority by developers than the usability experts.

Since we did not find any research on usability defect reporting in the literature, the relevant research investigating the influence of different factors on the software testing practices in general was helpful. We speculate the *role of the reporter, reporters' knowledge* and *experience*, *defect discovery methods* and usefulness of *automation tools* might have significant impact on reporting usability defects as well. To investigate these factors, we carry out a survey of software development practitioners from industry who have experience in dealing with usability defects. Closed-ended questions were used to gain respondents opinions and feedbacks. Participants participated voluntary in the survey. Key research questions we wanted to answer were:

- RQ1 - Is there significant difference in the way usability defects are described by the participants who have received usability/human computer interaction (HCI) training and those who do not?
- RQ2 - Is there significant difference in the way usability defects are described between participants who have experience in usability testing and those who do not?

- RQ3 - Is there significant difference in the way usability defects are described between the group of participants who used automated tools and those who do not?
- RQ4 - Is there significant difference in the way usability defects are described between participants who conduct usability testing and those who do not?

Section II gives a brief description of our survey and Section III presents the results in detail. Section IV analyses some possible threats to the validity of the survey and Section V discusses key findings. Finally, Section VI concludes the paper including directions for future research.

II. Study Design

A. Selection of Participants

We used a survey of software practitioners to collect their current practices, challenges and perspectives on usability defect reporting. Our participants were selected from software development practitioners with varying experience levels and roles (including developers, testers, and managers). The respondents were recruited from both open source and industrial communities. For open source respondents, we advertised the survey through the community forum, such as Eclipse Community forums. While industrial respondents were invited through Facebook, LinkedIn, Software Testing Club[1] and researchers' industrial contacts. Participation was voluntary and participants were allowed to discontinue participation at any time during the research activity. The consent to participate in the survey was implied by the return of the anonymous questionnaire.

B. Questionnaire Design

The survey had a total of 50 questions, divided in two parts investigating "usability defect attributes" and "factors that influence usability defect report quality". The questions included both open-ended and closed-ended questions. However, not all questions were answered by the participants. Only relevant questions are asked based on the participant's role (developers or reporters) in handling usability defects.

The first part explored what information do reporters use to describe usability defects and what information do developers consider useful for fixing usability defects. These findings have been published in a separate article [20].

In this paper, we only reported the second part of the survey findings. From our literature review, we listed key factors of *role of the reporter, reporters' knowledge* and *experience*, *defect discovery methods* and *automation tools* that we believed might influence the quality of usability defect report. The question used Likert scale with five levels of agreements ("Completely disagree", "Somewhat disagree", "Neither disagree or agree", "Somewhat agree" and "Completely agree"). Each factor consists of several questions as summarized below:

The role of the reporter: This factor had two closed questions. Q33 listed five statements to investigate the way technical and non-technical users report usability defects. The software developers were required to rate the frequency of these statements may be true when they assess a usability defect. Q45 asked both software developers and reporters whether they think the level of detail of usability defects may vary between reporters.

Knowledge: This factor had four closed questions. Q7 and Q8 asked respondents to indicate their participation in any usability-related training/ certification and rate the usefulness of the training respectively. Q41 asked respondents whether they think defect report form should be customized according to reporters' knowledge. Q46 listed seven knowledge factors that may affect the effectiveness of writing a good usability defect description. Respondents were required to indicate their level of agreement that these were important.

Experience: This factor had four closed questions that asked respondents' opinions on experience in usability testing (Q26 and Q34), and general software testing (Q48). Q40 asked respondents whether they think defect report form should be reflected to reporters' level of experience.

Automation tools: This factor had four questions. Q42 asked if reporters have experience in using any automation tools to capture usability defect information. In accompanying open question (Q43) reporters are requested to name the tools. Q44 asked reporters' opinions on the limitation of the manual process to capture usability defect information. Q39 asked whether respondents think user experience of using defect reporting tools may influence the quality of defect reports.

Defect discovery methods: This factor had two closed questions. Q11 listed fives statements on the availability of usability defect information for different defect discovery method, and respondents were required to indicate their agreement on these statements. Q12 asked whether respondents think that defect report form should be customized according to how usability defects are discovered.

The survey was piloted with Swinburne Software Innovation Lab (SSIL) software engineers and fifteen software developers recruited during a developer conference (DDD Melbourne 2014). Based on the verbal comments and the pattern of responses received, the survey instruments were refined. The full survey is available at http://bit.ly/UsabilityDefectsReportingSurvey. This survey study was approved on behalf of Swinburne's Human research Ethics Committee (SUHREC) by a delegated SUHREC subcommittee (SHESC2) (Approval number: SHR Project 2014/231).

C. Data Analysis

The Likert scales responses were converted to numerical values (5 implies "always", 4 implies "often", 3 implies "sometimes", 2 implies "rarely", and 1 implies "never") for analysis. Due to the low responses of scores 1 to 3, the 5-point Likert scores were recoded into seldom/ often/ always scores to compensate for potential response bias. The seldom score (1 to 3; never, rarely, sometimes) were converted to score of 1 to

[1] http://www.softwaretestingclub.com/forum

indicate “seldom”, and the “often” and “always” scores were retained. Chi-Square tests were used for statistical analysis. Since multiple Chi-Square are being performed simultaneously, we used the Bonferroni correction to avoid influence of spurious positives. In this case, we lowered the *P* value by dividing *P* = 0.05 by the number of tests to get the Bonferroni critical value. For example, to measure the association of usability defect attributes and usability/ HCI knowledge for reporters (attributes tested = 10) and software developers (attributes tested = 16) a test would have $P<0.05/10 = 0.005$ and $P<0.05/16 = 3.125\text{x}10^{-3}$ to be significant.

In addition, participants indicated their agreement about factors that influence usability defect reporting using five scale scores questions measuring the role of reporter, importance of knowledge and experience, influence of defect discovery methods, and usefulness of tools (1 implies “strongly disagree”, 2 implies “somewhat disagree”, 3 implies “neither disagree nor agree”, 4 implies “somewhat agree” and 5 implies “strongly agree”). Since each factor consists of several questions, composite scale scores were used for analysis by taking a mean of all the questions for each scale.

III. Findings

The data was collected during June – November 2015. A total of 294 respondents attempted the survey. However, only 147 responses were included in this analysis. The remaining 50% responses were excluded for no response beyond the first parts of the questionnaire. One possible explanation of the high percentage of invalid responses is due to the *out of scope* problems, where the respondents are not in the target population. For example, the software development practitioners who do not have experience in dealing with usability defects.

A. Demographic Information

The majority of the respondents were (65.3%) male, with 34.0% female participants, and 0.7% of participants who did not indicate their gender. About 85% of the respondents were between 25 - 44 years of age. The majority of the respondents are software developers (40.9%) followed by software testers (14.8%) and project managers (10.0%). For project managers, about 87% of them involve in defect reporting work. In terms of year of experiences, 63.8% of respondents had one to five years of work experience in their current position, while 25.3% had more than five years.

Most respondents indicated experience reporting usability defects (55.8%), while 44.2% had experience fixing them. The majority has more than 5 years of experience in software testing (34.1%). More than 60% of respondents indicated that they found usability defects when performing *exploratory testing*, *functional testing* and *using the product*. While a smaller proportion of respondents indicated that they discovered them through *alpha/beta testing* (26.8%). From the free-text explanation, some respondents explicitly mentioned other methods - *focus groups, GUI testing, performance testing, heuristics evaluation* and *automated testing*.

B. Factors Influencing Usability Defect Reporting

We asked respondents to indicate from a set of factors (“role of reporter”, “knowledge of specific usability/ software engineering”, “experience in software/usability testing and defect reporting”, “automation tools”, and “defect discovery methods”) and their influence on the presence of certain usability defect information in defect reports.

1) Role of the Reporter

We obtained developer feedback on the influence of role of reporter when they assessed and dealt with usability defects (Q33). As shown in Figure 1, more than 50% of software developers agreed that technical users, such as usability experts and developers provide more informative usability defect information, which include proposed solutions.

Reponses to Q45 suggest significant variability in level of detail between reporters (see Table 1).

TABLE I. Responses to “The level of detail of usability defect reports varies greatly from reporter to reporter” (Q45)

Completely disagree	0.7%
Somewhat disagree	4.1%
Neither disagree or agree	18.4%
Somewhat agree	29.9%
Completely agree	29.9%
No response	0.7%

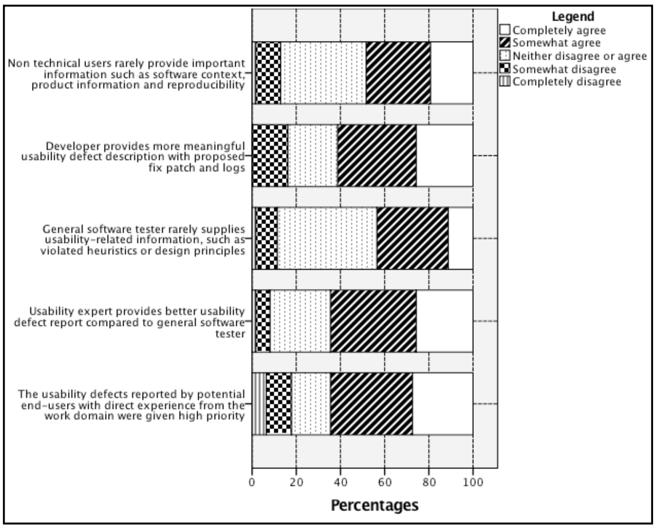

Fig. 1. Role of reporter in reporting usability defects (Q33)

2) Knowledge of Specific Usability/ Software Engineering

We listed seven knowledge factors (“practical knowledge”, “user’s perspective thinking”, “users’ mental model”, “usability principles”, “domain expertise”, “technical knowledge”, and “defect reporting”) that we postulated might influence the ability of respondents to write a good usability defect reports. More than 50% of respondents to the different factors agreed that these factors were influential (see Figure 2). The level of agreement is highest for “user’s perspective thinking” and “practical knowledge”.

In the related question (Q41), we obtained respondents opinion on the value of a “custom defect report form” – whether defect reporting tool should provide a custom form depending on reporter’s knowledge (non-technical, technical, HCI expert). While 51% agreed that defect report form should map to the reporter’s knowledge, 11.5% disagreed and 21.1% neither disagree nor agree (see Table 4). Only a minority (about 17%) indicated that they had received usability-related training/ certification (see Table 2). Of those, about 12% of them have experience in reporting usability defects and the remaining 5% fixing them. For those who had acquired the related training, the majority of them found the training were useful (see Table 3).

TABLE II. USABILITY-RELATED TRAINING RECEIVED (Q7)

Role in dealing with usability defects	Participation in usability-related training	
	Yes	No
Reporting usability defects	17	65
Fixing usability defects	8	57

TABLE III. “USEFULNESS” OF USABILITY-RELATED TRAINING (Q8)

Very useful	44%
Somewhat useful	40%
Neither useful or not useful	4%
Not very useful	12%
Complete waste of time	0%

TABLE IV. RESPONSES TO QUESTION “DO YOU THINK YOUR DEFECT REPORTING TOOL SHOULD PROVIDE CUSTOM FORMS FOR REPORTING DEFECTS DEPENDING ON REPORTER’S KNOWLEDGE?” (Q41)

Completely disagree	5.4%
Somewhat disagree	6.1%
Neither disagree or agree	21.1%
Somewhat agree	33.3%
Completely agree	17.7%
No response	16.3%

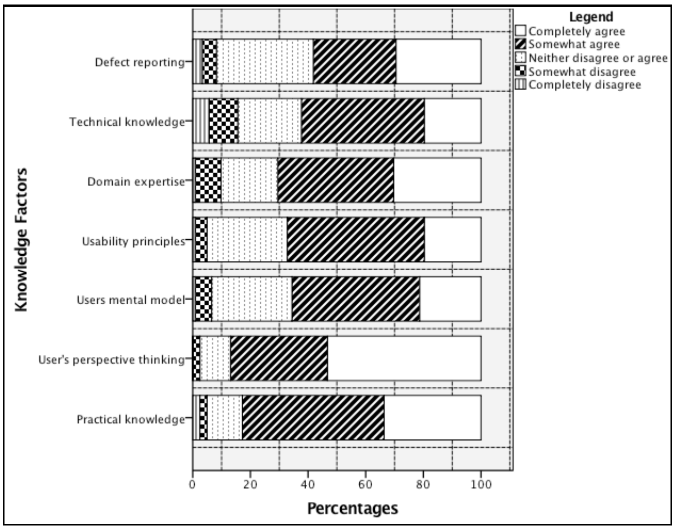

Fig. 2. Responses on “knowledge factors” in reporting usability defects (Q46)

To answer RQ1, a Chi-Square test was performed to examine the relationship between participants who received usability (or related) training and usability defect attributes. As can be seen in Table 5, three of the usability attributes are significant (severity, cause, observed) for reporters. As we tested 10 usability defect attributes, we expected one or two attributes to show a significant result purely by chance. By applying the Bonferroni correction, at significant level 0.005 ($P < 0.05/10 = 0.005$), only the test for severity ($\chi 2$, (2, N=82) = 13.725, $P < 0.005$) is significant. However, at significant level $P < 0.05/16 = 3.125 \text{ x } 10^{-3}$ (the number of usability attributes tested are 16 for software developers), there is no significant relationship between usability knowledge and usability defect attributes for the software developers.

Conclusion: Reporters who have usability and/ or HCI knowledge were likely to provide severity ratings in describing usability defects than those who do not have formal usability-related knowledge.

TABLE V. ASOCIATION BETWEEN USABILITY/ HCI KNOWLEDGE AND FREQUENCY OF USABILITY DEFECT ATTRIBUTES ARE SUPPLIED

Usability defect attributes		Software developers		Reporters	
		χ2 (df=2, N=65)	p-value	χ2 (df=2, N=82)	p-value
Title/ summary		1.741	0.419	3.692	0.158
Cause		0.832	0.660	6.820	0.033
Context		2.589	0.274	5.366	0.068
Obs		0.884	0.643	6.623	0.036
Exp		1.541	0.463	0.703	0.703
STR		0.518	0.772	5.732	0.057
Severity		1.783	0.410	13.725	0.001
Solution	Proposed solution	1.520	0.468	2.857	0.240
	Fix patch	0.333	0.847		
	Digital mockups	2.737	0.255		
	ASCII art	0.928	0.629		
SoftInfo	Product	1.332	0.514	4.773	0.092
	Component	5.208	0.074		
	Version	1.435	0.488		
TestEnv	Hardware	0.194	0.907	5.814	0.055
	Operating System	1.824	0.402		

3) *Experience in Software/ Usability Testing*

A majority of respondents agreed that experience in software testing and usability testing is a good influence on quality of report (~ 71% agreed in Q26). This view is supported by many software developers (~58.4%) that reporters are much better at proposing redesign solutions when they have usability experience (Q34). Furthermore, around 60% agreed, with minimal disagreement (less than 6%) that length of experience has a significant influence (Q48). In addition, nearly 62% respondents agreed that prior experience in usability defect reporting helps write better reports (Q49). Since no open-ended questions were asked around these factors, it is difficult to further clarify these findings.

We asked respondents whether they agreed that defect reporting tool should provide “custom defect report form” – that is defect report form should reflect the reporter’s experience, and their level of experience in software/ usability testing (Q40). Our findings show that considerably more respondents choosing “somewhat agree (31.3%)” than "completely agree (23.8%)", and only a small proportion disagree (12.9%).

To answer RQ2, the survey included questions about whether reporters had experience in software testing (no experience, less than 1 year, between 1 and 3 years, between 3 and 5 years or more than 5 years). The "less than 1 year", "between 1 and 3 years", "between 3 and 5 years" and "more than 5 years" scores were converted to a score of 1 to indicate "yes", and "no experience" scores were converted to a score of 0 to indicate "no". A Chi-Square test at significant P=0.05 was performed to examine the relationship between reporters with software testing experience and usability defect description attributes: there were seven significant attributes (see Table 6) However, by applying Bonferroni critical value, at significant level P=0.005, only relations between software testing experience and title/summary (χ2, (2, N=82) = 15.216, p < 0.005), context (χ2, (2, N=82) = 24.696, p < 0.005), expected results (χ2, (2, N=82) = 11.187, p < 0.005), and severity (χ2, (2, N=82) = 13.508, p < 0.005) were significantly observed.

Conclusion: Reporters with software testing experience were likely to provide title/ summary, context, expected results, and severity other than those who do not have software-testing experience.

TABLE VI. ASOCIATION BETWEEN SOFTWARE TESTING EXPERIENCE AND FREQUENCY OF USABILITY DEFECT ATTRIBUTES ARE SUPPLIED

Usability defect attributes	χ2 (df=2, N=82)	*P*-value
Title/ summary	15.216	0.000
Cause	4.963	0.084
Context	24.696	0.000
Solution	2.300	0.317
Obs	8.005	0.018
Exp	11.187	0.004
STR	9.058	0.011
Severity	13.508	0.001
SoftInfo	6.828	0.033
TestEnv	6.527	0.038

4) Influence of Defect Reporting and Automation Tools

Defect reporting tools to manage usability defects. In response to Q25 and Q35, more than 50% of the reporters and 38.7% of software developers indicated that they use defect reporting tools. Common tools (Q36) were *JIRA, Bugzilla* and *Redmine*. *Mantis, Trello, IBM Rational Team Concert,* and *Visual Studio TFS* were also listed multiple times. For JIRA, Bugzilla and Redmine users - 90% agreed to some extent that the defect reporting tool offers sufficient flexibility to capture and manage usability defects (Q37), but free-text feedback revealed considerable negative satisfaction (Q38). The following are representative: *"Most of the defect reporting tool do not have exhaustive options for usability defects"* and *"JIRA more customized by client but no specific customizations done for usability*".

Automation tools to capture usability defect information. In response to Q42, only 16% of reporters used tools to capture additional usability defect data, while 61% have never used any additional tools. For those who used automated tools, we asked them to name the tool, using an open-ended question (Q43). Among the tools mentioned were *CrazyEgg, Google Analytics, Microsoft Snipping tool, QuickTest Professional (QTP), WinRunner, Snagit, LICEcap, Screencast, Jing* and *Selenium*. We asked reporters to indicate whether they agreed that the manual process used to capture usability defect information is a cause of erroneous or incomplete defect reports (Q44). Our findings reported a mixed response, with roughly 20% of the reporters agreeing to some extent, but 22% disagreed and 35.4% neither agreed nor disagreed. Since, no open-ended question was provided, it is difficult to justify the satisfaction and feedback of those tools. It is notable that a high proportion of respondents either indicated no view, or did not answer the question at all (23.2%).

In a related question, we asked respondents' opinion if the user experience of using defect reporting tools might influence the quality of defect reports (Q39). Our findings indicate, nearly 56% believed that user experience is one of the key considerations when reporting usability defects.

To answer RQ3, a Chi-Square test was performed to examine the relation between reporters that used automated tools to capture usability defects and usability defect description attributes. At Bonferroni correction significant level P=0.005, a significant association exists between used of automated tools and context (χ2, (2, N=63) = 11.443, P< 0.005). Table 7 summarizes the statistical results.

Conclusion: Reporters who used automated tools were likely to capture software context better than other those who used manual process. This is likely to be a reflection of the underlying maturity of the team due to their selection of automation, rather than a direct influencer.

TABLE VII. ASOCIATION BETWEEN AUTOMATED TOOLS AND FREQUENCY OF USABILITY DEFECT ATTRIBUTES ARE SUPPLIED

Usability defect attributes	χ2 (df=2, N=63)	*P*-value
Title/ summary	5.418	0.067
Cause	1.079	0.583
Context	11.443	0.003
Solution	4.435	0.109
Obs	4.199	0.123
Exp	3.311	0.191
STR	3.482	0.175
Severity	1.920	0.383
SoftInfo	0.536	0.765
TestEnv	5.054	0.080

5) Influence of Defect Discovery Methods

The vast majority of respondents agreed that the amount of information available for reporting usability defects varies according to how the defects are discovered. In this context, defects found during usability testing reveal more information as compared to other types of testing – approximately 87% of respondents agreed that using usability testing information about user's knowledge, likely difficulties, actual task scenario and realistic redesign solutions can be obtained from the actual user, and approximately 73% agreed other usability-related information such as violated heuristics or design principles, and user response and feelings can be collected (see Figure 3).

In accompanying question (Q12), we asked respondents whether defect reporting tool should provide “custom defect report form” (that is defect report form should be designed to map to different types of defects) - majority of the respondents (~63.9%) agreed and only 7.5% disagreeing on this idea.

To answer RQ4, the survey asked questions how reporters discover usability defects (i.e., exploratory testing, functional testing, usability testing, beta/ alpha testing, complaints/ reports from customers, using the product, and other). Participants were allowed to choose multiple answers. Any response that consists of usability testing option were converted to “usability testing” response options, while other responses that do not have usability testing option were converted to “not using usability” response options. A Chi-Square test was performed to examine the relation between reporters that conducted usability testing and usability defect description attributes. Table 8 shows that at significant level 0.05 three attributes were significant, however by applying Bonferroni correction a test would have to have P<0.005 to be significant. Therefore, only the cause is significant ($\chi 2$, (2, *N*=82) = 14.530, p < 0.005).

Conclusion: Reporters who discover usability defects using usability test were likely to provide possible cause other than those who used other testing methods – this finding reinforces the value of usability testing.

TABLE VIII. ASOCIATION BETWEEN USABILITY TESTING AND FREQUENCY OF USABILITY DEFECT ATTRIBUTES ARE SUPPLIED

Usability defect attributes	**χ2 (df=2, N=82)**	***P*-value**
Title/ summary	2.045	0.360
Cause	14.530	0.001
Context	4.361	0.113
Solution	1.146	0.564
Obs	6.066	0.048
Exp	3.245	0.197
STR	6.020	0.049
Severity	5.274	0.072
SoftInfo	4.455	0.108
TestEnv	0.270	0.874

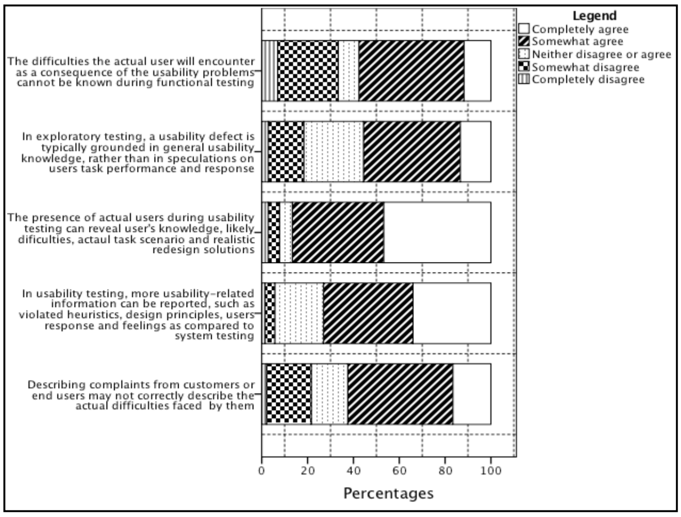

Fig. 3. “Influence of defect discovery methods” in reporting defects (Q11)

IV. THREATS TO VALIDITY

Internal Validity - The main threat to this study is a misunderstanding of the survey context by the respondents. Our goal is to focus on usability defect reporting instead of general software defect reporting. Respondents may have answered the questionnaire based on their general defect reporting knowledge and experience. We addressed this threat by (a) giving three different types of usability defect examples at the beginning of the survey, (b) highlighting the ***usability defects*** (bold and italic) keyword for every question, so the respondents were always aware of the survey context.

External Validity - One possible external threat to the validity of the survey outcome is the representativeness of the respondents. Although 147 valid responses to the survey is a good start, the population size is relatively small to examine the research questions and answer them in appropriate manner. There could have been some level of confirmation bias among the participants as not all of them were directly working in HCI projects, reporting usability defects, or even using formal defect reporting processes and tools. For example, only a small number of people used automation tools (16%) to report usability defects. As such, we cannot claim our findings about the question on automation tools to be conclusive. Instead, our survey may be a plausible start for more detailed empirical studies. Furthermore, the used of self-rating technique to identify how participants discover usability defects could not really reveal the participant’s experience, approach and skill in usability defect discovery.

Construct Validity - One concern is regarding incorrect measures, i.e. not precisely measuring respondents’ practices in reporting usability defects. To mitigate this concern, we reused previous surveys and added questions from both usability and software engineering fields. Another possible threat is that our respondent recommendation does not entirely reflect the true reality of defect reporting practice. Since our survey is anonymous, some responses we received stated that they have never used defect reporting tools. In fact, some comments are not meaningful.

V. DISCUSSION

A. The Role of the Reporter

The results of our survey suggest that - the level of detail of usability defect description varies considerably. At least 60% of respondents indicated that the reporters’ role matters a great deal in getting a usability defect fixed. From the software developers’ point of view, technical users often provided more informative usability defect information, such as redesign solutions. These findings do not mean, however, that software developers did not appreciate defect reports submitted by non-technical users, especially when they are clearly written, detailed and technically complete. Nevertheless, it is unfortunate that our survey did not investigate reporters’ background. As Zimmermann et al. [1] claimed, “well known reporters usually get more consideration than unknown reporters, assuming the reporter has a pretty good history in bug reporting”, this suggests that when software developers have put their trust in a reporter, they are likely to give it a higher priority. It is therefore, worthy of a targeted further examination. We also found that usability defects reported by customers typically get reviewed and fixed faster. Possibly, management tends to prioritize these defects.

B. The Role of Experience in Usability Defect Reporting

There are three categories of experience that affect the effectiveness of writing usability defect descriptions. First, experience in usability testing. The majority of respondents explicitly mentioned that usability testing experience is a valuable skill to create better usability defect reports. Second, experience in general software testing. About 60% of the respondents believed that experienced reporters generally provide more organized and detailed reports, however in the context of describing usability defects, this ability is uncertain. In practical usability work, defect reports with usability information are necessary to describe how the problems should be addressed [21], and the root cause [20]. However, we found that even experienced reporters seldom include solution proposal. In fact, in general software defect reporting, only a small group of experienced reporters could produce a good quality of report because of their limited understanding of what constituted a defect [22]. This suggests that even though software testing experience is useful for test execution and evaluation, it does not guarantee a high quality defect report. Third, experience in reporting usability defects. In general, our respondents tend to believe that previous experience in reporting usability defects is helpful to write better usability defect descriptions.

C. The Role of Knowledge in Usability Defect Reporting

Experience is often associated with deep knowledge. From the responses to the influential knowledge of usability defect reporting, it is apparent that “users’ perspective” and “practical knowledge” are important. As Itkonen and Lassenius [14] reported, a good understanding of usage procedures and context allows reporters to identify actual usability defects by reflecting on the system’s behavior with realistic usage tasks and context rather than based on personal speculation. As well as domain knowledge, some of our respondents agreed that understanding user’s mental model and usability principles are desired. However, only 17% respondents had received usability-related training. For respondents with related training, they seem to understand usability defects, and tend to provide severity ratings. Despite the value of this type of training it does not seem to be common. We assume usability-related training is normally conducted in the context of learning about user experience, building skills on UX best practices, and designing user interface[2], however the training seems to provide insufficient focus on writing good defect reports. If they are systematically taught this, they may not know the reporting tools as well.

Knowledge on defect reporting tools assist in writing good usability defect reports. Many respondents suggested that understanding reporting tool is important. As Ko and Chilana reported, lack of knowledge of defect reporting tool and process may influence the quality of defect reports [22]. Therefore, the familiarity of using defect reporting tools can help, however, unnecessary options in defect report forms were sometimes not relevant for all types of defects. The generic defect report forms in popular defect reporting tools were insufficient to construct comprehensive usability defect description [12], and often there are no specific customizations done for usability.

D. Use of Automation Tools

Not many reporters used automation tools to capture additional usability defect information. However, our findings show that the value of capturing additional context. Instead of knowing which particular interface element that cause confusion, it is more useful to know why users get confused [10]. Among the tools mentioned by our respondents to record screenshots, video and audio are Snagit, Screencast, Jing, LICEcap, and Snipping tool. Since these tools are not tightly integrated into reporting workflow, there is friction in creating useful hypermedia attachments.

Furthermore, software engineering research on collecting logs and traces [9], [23] and automatically capturing steps to reproduce [24], [25] does not seem to have a bigger contribution in practical usability testing context. Possibly, this preference for commercially available software may be attributed to the fact that tools developed by research community tend to be concept demonstrators with ad-hoc support.

E. Defect Information Obtained Through Usability Testing

In our survey, about 60% of reporters used usability testing to discover usability defects. The majority believes that usability information such as user’s knowledge, likely difficulties, actual task scenario, realistic redesign solutions, violated heuristics, and users feelings are more easily obtained via usability tests. In addition, we also found reporters who conducted usability test are able to explain possible causes better other than traditional testing. Possibly, when reporters observed users performing certain tasks, they gain direct feedback from the users. Quantitative data such as users’ performance (i.e., time on task and error rates) also can be collected. In many cases, usability defects are not considered high severity [26] that cause harm either directly or financially. By supplying additional information, software developers may address usability defects more aggressively.

VI. Conclusion

We conducted a survey of nearly 150 software practitioners to examine the factors that influence quality usability defect reporting. Our findings increase understanding of usability defect reporting from the viewpoint of the effects the knowledge and experience have on testers. However, our finding is preliminary and need to be extended. Researchers could explore the effect of implicit usability testing activities, and the type of knowledge and experience applied when reporting usability defects. Key future investigations we plan to carry out include:

Identify important usability defect attributes – usability defects tend to get less priority compared to functional defects. One possible reason may be that developers do not understand

[2] https://www.nngroup.com/training/

the problem or consider it as valid. We could define and capture specific usability defect attributes to reveal more usability characteristics, such as violated heuristics, interaction difficulties, and user's feeling and emotion so usability defects can be better understood and appreciated by developers.

Support lightweight defect reporting – generic defect reports forms are offered by most defect reporting tools and do not consider the reporters' knowledge/ experience, and discovery method. A range of enhancements to existing defect reporting tools are suggested [9], [23], [24], [25] but none of them are specifically designed to support customization usability defects. Novices in particular struggle to prepare good usability defect reports [27]. Using guided reporting should great assist – for example, reporters can be assisted with predefined attributes for input selection, online help, and question/ wizard-based interaction. Defect report forms can also be customized to reflect reporter's profile (i.e., non-technical user, usability expert, customer, etc).

Integration of automation tools and reporting tools – many usability engineering tools offer automated data capture and richer kinds of information capture e.g., instrumenting applications to capture traces and user interaction, recording richer user interaction and mapping to user task, and capture of video, audio, screenshots, diverse interaction (touch, sketch, gesture, accelerometer, as well as keyboard and mouse). However, most software defect reporting tools make capture of this highly manual, uni-format (usually free format text), or make adding and manipulating attachments difficult. Combining HCI usability engineering methods and tools with software defect reporting and management repositories will be valuable.

Acknowledgment

Support for the first author from the Ministry of Higher Education Malaysia, Universiti Teknologi MARA (UiTM), and from the Swinburne Software Innovation Lab and the National ICT Australia for all authors, is gratefully acknowledges.

References

[1] T. Zimmermann, R. Premraj, N. Bettenburg, C. Weiss, S. Just, and A. Schro, "What Makes a Good Bug Report ?," *IEEE Trans. Softw. Eng.*, vol. 36, no. 5, pp. 618–643, 2010.

[2] E. I. Laukkanen and M. V. Mantyla, "Survey Reproduction of Defect Reporting in Industrial Software Development," in *2011 International Symposium on Empirical Software Engineering and Measurement*, 2011, pp. 197–206.

[3] T. Kanij, R. Merkel, and J. Grundy, "A preliminary survey of factors affecting software testers," in *Proceedings of the Australian Software Engineering Conference, ASWEC*, 2014, pp. 180–189.

[4] S. Breu and J. Sillito, "Information Needs in Bug Reports : Improving Cooperation Between Developers and Users," in *The 2010 ACM Conference on Computer Supported Cooperative Work*, 2010, pp. 301–310.

[5] C. Camacho, S. Marczak, and T. Conte, "On the Identification of Best Practices for Improving the Efficiency of Testing Activities in Distributed Software Projects: Preliminary Findings from an Empirical Study," *2013 IEEE 8th Int. Conf. Glob. Softw. Eng. Work.*, pp. 1–4, Aug. 2013.

[6] J. Li, S. Tor, R. Conradi, and K. J. M.W., "Enhancing Defect Tracking Systems to Facilitate Software Quality Improvement," *IEEE Softw.*, vol. 59–66, 2012.

[7] T. Zimmermann and S. Breu, "Improving Bug Tracking Systems," in *31st International Conference on Software Engineering - Companion Volume, 2009. ICSE-Companion 2009.*, 2009, pp. 247 – 250.

[8] B. Dit and A. Marcus, "Improving the Readability of Defect Reports," in *Proceedings of the 2008 International Workshop on Recommendation System for Software Engineering*, 2008, pp. 47–49.

[9] S. Herbold, J. Grabowski, S. Waack, and U. Bünting, "Improved Bug Reporting and Reproduction through Non-intrusive GUI Usage Monitoring and Automated Replaying," in *2011 IEEE Fourth International Conference on Software Testing, Verification and Validation Workshops*, 2011, pp. 232–241.

[10] D. M. Nichols and M. B. Twidale, "Usability processes in open source projects," *Softw. Process Improv. Pract.*, vol. 11, no. 2, pp. 149–162, Mar. 2006.

[11] L. Zhao and F. P. Deek, "Improving Open Source Software Usability," in *Proceeedings of the Eleventh Americas Conference on Information Systems*, 2005.

[12] N. S. M. Yusop, J. Grundy, and R. Vasa, "Reporting Usability Defects : Limitations of Open Source Defect Repositories and Suggestions for Improvement," in *Proceedings of the ASWEC 2015 24th Australasian Software Engineering Conference*, 2015, pp. 38–43.

[13] J. Itkonen and K. Rautiainen, "Exploratory testing: a multiple case study," in *IEEE/ACM International Symposium on Empirical Software Engineering and Measurement - ESEM*, 2005, vol. 00, pp. 82–91.

[14] J. Itkonen and C. Lassenius, "The Role of the Tester ' s Knowledge in Exploratory Software Testing," *IEEE Trans. Softw. Eng.*, vol. 39, no. 5, pp. 707–724, 2013.

[15] C. R. L. Neto and E. S. de Almeida, "Five years of lessons learned from the Software Engineering course: Adapting best practices for Distributed Software Development," *2012 Second Int. Work. Collab. Teach. Glob. Distrib. Softw. Dev.*, pp. 6–10, Jun. 2012.

[16] V. Kettunen, J. Kasurinen, O. Taipale, and K. Smolander, "A study on agility and testing processes in software organizations," *Hum. Factors*, pp. 231–240, 2010.

[17] A. Beer, A.- Vienna, and R. Ramler, "The Role of Experience in Software Testing Practice," in *34th Euromicro Conference Software Engineering and Advanced Applications, 2008. SEAA '08.*, 2008, pp. 258–265.

[18] P. L. Poon, T. H. Tse, S. F. Tang, and F. C. Kuo, "Contributions of tester experience and a checklist guideline to the identification of categories and choices for software testing," *Softw. Qual. J.*, vol. 19, pp. 141–163, 2011.

[19] A. F⊘lstad, "Work-Domain Experts as Evaluators: Usability Inspection of Domain-Specific Work-Support Systems," *Int. J. Hum. Comput. Interact.*, vol. 22, pp. 217–245, 2007.

[20] N. S. M. Yusop, J. Grundy, and R. Vasa, "Reporting Usability Defects – Do Reporters Report What Software Developers Need ?," in *Proceedings of the 20th International Conference on Evaluation and Assessment in Software Engineering*, 2016.

[21] K. Hornbæk and E. Frokjær, "Comparing usability problems and redesign proposals as input to practical systems development," in *CHI 2005: Technology, Safety, Community: Conference Proceedings - Conference on Human Factors in Computing Systems*, 2005, pp. 391–400.

[22] A. J. Ko and P. K. Chilana, "How power users help and hinder open bug reporting," in *Proceedings of the 28th international conference on Human factors in computing systems - CHI '10*, 2010, p. 1665.

[23] T. Roehm, N. Gurbanova, B. Bruegge, C. Joubert, and W. Maalej, "Monitoring user interactions for supporting failure reproduction," in *2013 21st International Conference on Program Comprehension (ICPC)*, 2013, pp. 73–82.

[24] J. Pichler and R. Ramler, "How to test the intangible properties of graphical user interfaces?," in *Proceedings of the 1st International Conference on Software Testing, Verification and Validation, ICST 2008*, 2008, pp. 494–497.

[25] K. Moran, M. Linares-Vásquez, C. Bernal-Cárdenas, and D. Poshyvanyk, "Auto-completing Bug Reports for Android Applications," *Proc. 2015 10th Jt. Meet. Found. Softw. Eng.*, pp. 673–686, 2015.

[26] C. Wilson and K. P. Coyne, "The whiteboard: Tracking usability issues: to bug or not to bug?," *Interactions*, pp. 15–19, 2001.

[27] V. Arnicane, *Use of Non-IT Testers in Software Development*, vol. 4589. 2007, pp. 175–187.