Testing Touch: Emulators vs. Devices

Rebecca Baker, Xiaoning Sun, and Bob Hendrich

CA Technologies One CA Plaza
Islandia, NY 11749, USA
rebecca.baker@ca.com, xiaoning.sun@ca.com,
bob.hendrich@ca.com

Abstract. Conducting usability tests is a time and resource intensive process. The ability to do remote testing significantly reduces the cost associated with testing while still providing rich feedback. However, mobile touch interfaces provide a unique challenge for remote testers. This paper compares the results of testing an application on a mobile touch device (the iPod Touch) and an emulator using a between subject test. In it, we examine the differences in time on task, usability issues discovered, and task completion. Results suggest that emulators can be used to detect some but not all usability issues and that they may provide some false positives as well. Further research is required to separate issues with the emulator vs. device from the remote vs. in-person test environment.

Keywords: Touch Interface, Emulator**,** Remote Usability, Usability Test, User-Centered Design.

1 Introduction

Mobile touch interfaces are daily becoming more ubiquitous. In addition to the popular uses of such devices as entertainment and social networking supports, many large software firms are delving into the mobile market. These companies are developing applications that provide extensions to enterprise software, allowing customers access to functionality that had previously required them to be on-site. As these applications tend to be complex, usability testing becomes more important.

Remote usability tests are often the mainstay of corporate testing. Getting feedback on a global level has become paramount with the worldwide market pursued by most large software companies. Getting feedback from an international audience early in the process reduces costs and ensures a universally-usable product design. This is especially true of mobile application development, where guidelines are still being developed and the interaction challenges are not well-understood.

Unfortunately, remote testing with mobile applications is problematic at best. Short of shipping the device to the end user and asking them to film themselves using the application, user researchers must rely on emulators which simulate the mobile experience on the subject's computer. Studies looking at traditional remote usability tests [1], [2] suggest that significant differences between remote usability tests (both synchronous and asynchronous) and traditional tests do not exist, in terms of the

P.L.P. Rau (Ed.): Internationalization, Design, HCII 2011, LNCS 6775, pp. 135–142, 2011.

number of usability issues found, their types, or their severities. But using an emulator is not an identical experience to using a mobile application, especially one with a touch interface. The emulator cannot be manipulated like the mobile device and the required use of the mouse adds another layer of difference to the experience. Previous in-person studies [3] suggest that this difference is not significant when detecting usability errors.

This experiment seeks to understand the differences in results from a remote test done with an emulator and an in-person test done with a physical device, specifically the iPod Touch®.

2 Methodology

We chose a product for our study that is currently available through the iTunes app store, but sufficiently specialized and technical that the majority of test subjects would not have seen it. The application lets users track and predict the location of satellites in the sky. Primarily used by satellite watchers and ham radio operators, the application was developed by an enterprise software designer. The application was ideal for this experiment as it had some obvious usability issues that the designer had not had a chance to evaluate and correct.

Fig. 1. ProSat Application

We evaluated the product using standard usability heuristics and chose tasks for the test that would highlight these issues.

2.1 Participants

All participants were CA Technologies employees with normal or corrected-to-normal vision. A total of 49 users were tested, 24 remotely and 25 in-person. For the in-person tests, all subjects were located in either Colorado Springs, CO or Plano, TX. For remote tests, subjects were located in numerous locations around North America.

The demographic breakdown of the participants was as follows:

Table 1. Gender

	Male	**Female**
Gender (emulator)	18	6
Gender (device)	16	9

Table 2. Age

	21-30	**31-40**	**41-50**	**51-60**	**61-up**
Age (emulator)	0	6	15	2	1
Age (device)	0	3	14	5	3

Table 3. Frequency of use of touch device

	Never	**Once or Twice**	**Monthly**	**Weekly**	**Daily**
Frequency of use (emulator)	0	3	2	3	16
Frequency of use (device)	2	7	0	0	16

2.2 User Preparation

When they arrived at the test, all users were given a brief description of the purpose of the test, a general description of the software and its uses, and the format of the test. They were provided an opportunity to ask questions both before and after the test.

In addition, before the test, remote users received an instruction email that provided them the time and date of their test. They were asked to have the following items ready before the test started:

- Access to a speaker phone or a headset, with toll-free access. Because the test required the use of a mouse and keyboard, we asked that the users have a way to communicate with us without encumbering their hands.

- A printed copy of the task list. We wanted the users to access the software in a full screen mode, which made it necessary for them to have a printed copy of the tasks.

2.3 Test Environment

All subjects were presented with the same tasks in the same order. In-person tests were performed in a team room with standard office seating (fixed height table and chairs) and overhead fluorescent lights. Remote tests were performed at the subject's individual location which could vary significantly. Remote users accessed the software using Microsoft LiveMeeting. The test then proceeded like a traditional usability test.

During the test, users performed the following eight tasks:

1. Rotate the globe to see North America.
2. Zoom in on the ISS (ZARYA) Satellite.
3. View the current details of the ISS Satellite. Change the base frequency of RCV for the ISS Satellite to 5. View the latitude/longitude data for the ISS Satellite for the next 7 days.
4. Add the COSMOS Satellite, under Amateur Satellites (CelesTrak), to the satellite list. Change the COSMOS Satellite color to blue and set Number of Orbits to Draw to 2. Remove the COSMOS Satellite that you just added from the satellite list.
5. Find which disaster monitoring satellites can be viewed from your location during the next hour.
6. Find the Sun rise/set information for tomorrow. Make the Sun rise/set menu display as the second icon in the tool bar on the bottom.
7. Change the system time within the application to be 7PM today.
8. Find information on how to set viewing information from the help document.

In addition, subjects were asked to rate the usability of the interface on a Likert scale of 1 to 7 and were asked open-ended questions about their experience with touch interfaces.

We found the remote setup had very poor response time. This was largely due to the subjects having to interact with the emulator through Microsoft LiveMeeting.

2.4 Measurements and Analyses

A number of metrics were measured in this study, such as number of errors, number of assists, time taken for completing each task, and task completion. Errors are defined as any action, either intentional or unintentional, that adds to, deletes from, modifies, or otherwise affects the systems, or a failure to save changes. An assist is any help given to the user by the test administrator that helps the user perform the task. If a user committed three errors or needed three assists for a given task, the task was marked as not completed.

The analyses were carried out by Analyses of Variance and t-tests.

3 Results

We found significant testing environment effects on time and errors. Users in the device group made more errors (F (1, 47) = 7.81, $p < .01$). In addition, it took users in the emulator group more time to complete tasks (F (1, 47) = 51.46, $p < .00$).

Table 4. Mean (SD) of time, assists and errors, and p-values from ANOVA analyses

	Device N=25	Emulator N=24	P-value
Time	102.62 (35.96)	177.21 (36.82)	.00
Assists	0.37 (0.24)	0.47 (0.31)	.19
Errors	0.13 (0.12)	0.47 (0.81)	.01

Fig. 2 shows age effects on assists. We found a relationship between the number of assists and the age of the user, in which older users requested assistance more frequently (F (3, 45) = 5.77, $p < .00$). The post-hoc Tukey test revealed a significant difference between the 31-40 age group and the 61 and up age group ($p < .00$) and a significant difference between the 31-50 age group and the 61 and up age group ($p < .00$). There was no significant difference between the 51-60 age group and the 61 and up age group. This finding was independent of the environment (device or emulator).

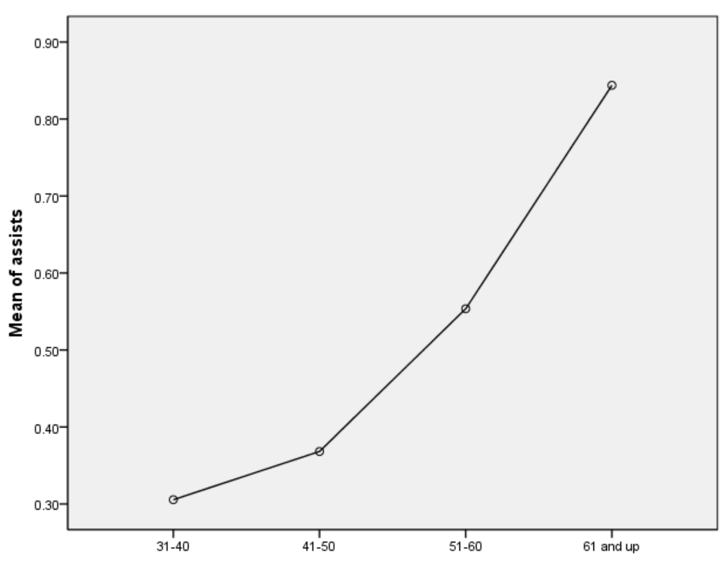

Fig. 2. Age affect on assists

Finally, we found the abandonment rate (users deciding to abandon the task rather than try to finish with or without help) was significantly higher on the emulator tests. Nine out of 25 users on the emulator abandoned Task 8, rather than finish it. None of the device users abandoned this task.

4 Discussion

The difference in results between the emulator and device are interesting, if not entirely surprising. The difference in time on task is easily explained by the significantly poorer response time for the emulator. However, the difference in error detection is significant. Subjects were more likely to find and detect errors on the device than through the emulator. Generally, we found the emulator participants were more likely to overlook issues with the arrangement of interface elements (see following figure) and visibility of features.

Fig. 3. Number of orbits to draw error

In Fig. 3, the field, Number of orbits to draw, is slightly off the bottom of the screen by default. Device users frequently tried to touch this field to enter a number but selected the Settings toolbar button instead. Emulator users did not have this accuracy issue due to use of a mouse and keyboard, rather than fingers.

The abandonment issue was also interesting. It was unclear from this study what caused emulator participants to be more willing to give up on a task. The response time was significantly worse for the emulator and may have caused enough frustration to cause users to be more willing to give up. However, the lack of social pressure (test administrator over the phone rather than in the room) or the lack of novelty (most device users commented that using a touch device is "fun") could also have contributed to this finding. Future research will have to be undertaken to gain a better understanding of this effect.

5 Conclusions

This study showed some limitations of remote testing for mobile touch interfaces, outlining the unique challenges currently faced in this area. While some results may be gathered using emulators, it is not an adequate substitute for in-person testing at this time. This finding is particularly interesting given the large number of developers who currently use emulators to test their interfaces during the development process. Tests run by the Nielsen group noted some common design issues for touch screens, such as 1) fat fingers- accidental tapping and the back button; 2) invisible controls; 3) getting lost in an application, i.e., lack of navigational aids; 4) inconsistent interaction design [4]. Certainly, more research should be done into how to improve the development environment as well as the test environment to prevent usability issues before they arise.

The marked difference between these findings and those of previous studies regarding remote usability testing suggests that traditional usability testing techniques, and perhaps even usability heuristics, need to be re-evaluated when approaching touch interfaces. Some work has already begun in this area [5] recognizing the differences in design requirements for mobile devices.

The abandonment issue provides an interesting mystery. It is uncertain from this experiment if the higher rate of abandonment on the emulator was due to the slow response time of the environment, the remote nature of the test, the lack of "cool" factor when compared to the actual touch device, a combination of these factors, or other factors entirely. Further research is needed to answer these questions.

Similarly, more research into the increased number of assists, as related to participant age, could provide significant insight into the effects of age on usability testing. With an aging IT workforce, the results of this type of research could take on new significance in software and interface design.

Acknowledgments. We would like to thank Craig Vosburgh for providing the application for this study and the CA Technologies, Inc. employees who generously volunteered their time for this study.

References

[1] Brush, A., Ames, M., Davis, J.: A Comparison of Synchronous Remote and Local Usability Studies for an Expert Interface. In: Proceedings of CHI 2004 Conference on Human Factors in Computing Systems, pp. 1179–1182 (2004)
[2] Hartson, H.R., Castillo, J.C., Kelso, J., Neale, W.C.: Remote Evaluation: The Network as an Extension of the Usability Laboratory. In: Proceedings of CHI 1996 Conference on Human Factors in Computing Systems, pp. 228–235 (1996)
[3] Betiol, A.H., Cybus, W.d.A.: Usability Testing of Mobile Devices: A Comparison of Three Approaches. In: HCI Interact IFIP TC13 International Conference, pp. 470–481 (2005)
[4] Budiu, R., Nielsen, J.: iPad Usability: First Findings From User Testing. Nielsen Norman Group Report (2010), http://www.nngroup.com/reports/mobile/ipad/ipad-usability.pdf
[5] Griggs, K., Bridges, L.M., Rempel, H.G.: Library/mobile: Tips on Designing and Developing Mobile Web Sites. Code4Lib Journal 8, 11–23 (2009)