Paper 142-30

Usability and Usability Testing at SAS

Paul Hankey, SAS Institute Inc., Cary, NC

ABSTRACT

This paper discusses the effects of usability testing on the design and quality of SAS® products. The first section gives an overview of usability testing, explains the place of usability testing in the development cycle, and discusses the roles that various individuals play in a usability test. The second section contains examples, including video highlights from SAS usability tests, to illustrate various concepts of usability testing.

INTRODUCTION

The word "usability" is tossed about quite a bit in the software design world. Even in usability newsgroups, debates continue as to the correct definition of usability. One of the more succinct and complete definitions of usability was formulated by the International Standards Organization (ISO), which defines usability as:

> The effectiveness, efficiency and satisfaction with which specified users achieve specified goals in particular environments.

DEFINING USABILITY

Several years ago, I performed a competitive usability test of two products. The products that were evaluated were designed to back up computers on a network. Both products had been on the market for several releases and had comparable, mature feature sets. The results of this test are used here to illustrate the ISO definition of usability.

- **To achieve specific *goals***
  The first task in the evaluation presented the participants with a very clear goal. The participants were asked to take the software out of the box, load it onto a PC, and use the interface to back up a workstation that was located on an adjacent table.

- **By specified *users***
  Sixteen IT administrators participated in the test. All of them had previous experience using network backup products.

- **In particular *environments***
  For this test, the environment was a usability laboratory. Normally, it would have been a computer room, an office, or someone's cubicle. The key is that the products were designed for use in a controlled environment.

- **As judged by *measures* of effectiveness, efficiency, and satisfaction**
  For Product A, 14 of the 16 participants were able to install and perform the backup in an average of a little more than 35 minutes. For Product B, 14 of the 16 participants *failed* the task. Of the 2 participants who were able to complete the installation and backup, 1 participant was stopped after working for over 9 hours on the task!

So, there was a very big difference in both the effectiveness and efficiency measures. You can only imagine how disparate the satisfaction scores were for these two products.

GAUGING USABILITY

Just as there are many definitions of usability, there are many ways of gauging whether a product is usable. Saying that a product is usable does not provide much information to the purchaser. One way to help solve the problem is to narrow the focus to various aspects of product use. For example, **a product is usable if it is:**

- ***Easy to install and configure***—Can the installer get the product up-and-running in an hour, or will it take a month to do it?

- ***Easy to learn***—How long does it take a user to become productive?

- ***Easy to use to perform daily tasks***—Wizards are designed to help users work through difficult processes; however, daily use of a wizard will quickly become cumbersome. Shortcuts should be considered for frequently performed tasks.

- ***Easy to recover from errors***—One of the most common mistakes in wording an error message is telling users that something is wrong, but not telling them how to correct it.

- ***Easy to maintain and update***—Does the current operating environment under which the product is run need to be taken down during maintenance or can it be kept running? Will product updates write over personal preferences?

WHAT ARE THE BENEFITS OF USABILITY TESTING?

The most obvious benefit of performing a usability test is finding problems before the product is made available to customers. If testing occurs early enough in the design cycle, complex issues can be corrected. However, if testing isn't performed until late in the design, development might not be able to make corrections before the product is shipped. The problem is thus passed on to the customer (and to technical support!).

One of the main rules of usability testing is to test early and test often. The earlier you can get in on the design cycle and the more often that you test, the better the product will be when it is released.

You should always try to test with your target user population if possible. Testing performed by customers gives you the best feedback. However, it can sometimes be difficult to get access to customers for this purpose.

Beware! You and your development team might not (and usually do not) have the same level of knowledge and experience as your real users. Something that might be clear to your team might be meaningless to your real users.

It is always a good idea to test the product when it reaches the general-availability stage. This represents a test of the actual product that the majority of your customers will use. While it is definitely too late to correct any major issues at this point, testing gives you information that can be used to plan work for the next release.

A competitive usability test with the general-availability product is also a great idea. Comparing your product against one or more primary competitors will provide a clear understanding of how competitive your product will be in the market. What are the strengths of your product? What are the deficiencies? Marketers love data from competitive usability tests. Particularly if your product does well, the data can be used to help position the sale of the product.

TYPES OF USABILITY TESTS

This section briefly describes some of the different types of usability tests.

PROTOTYPE TESTS

Optimally, the entire product should be prototyped and tested before coding begins. Begin testing with paper prototypes and move to semi-functional and fully functional prototypes for later phases of the design.

In reality, completely designing and testing a prototype is not possible for many complex software applications. The UI design for a complex product typically takes months to complete, so some development activities usually begin while the product is still being designed. This should not stop you from running multiple evaluations of the various components.

GENERAL AVAILABILITY/BASELINE TESTS

These tests help you plan for the next release and provide solutions to Technical Support so that the answers are ready when customers call. The tests also generate valuable data for use by marketing and sales.

COMPETITIVE TESTS

These tests compare a finished product against one or more competitors. Because users are exposed to all products, it is essential that the presentation of the products be counter-balanced (in the example of the network backup software that was given earlier, half of the users installed Product A first, and half of the users installed Product B first). Features are easy to compare on paper, but actually watching someone try to perform the same task on two different products yields compelling information—positive or negative.

OUT-OF-THE-BOX TESTS

An out-of-the-box usability test is just what it sounds like. The user is given a product in whatever package it is shipped in and is asked to get the product up-and-running. This is a test of the product packaging, the documentation, and the installation instructions.

FORMAL LAB TESTS

The name ”Formal lab tests” refers not only to the place that the test is conducted but to the methodology, as well. Formal lab tests “level” the playing field for all participants so that each participant is presented with essentially the same experience.

REMOTE TESTS

With the advent of high-speed Internet connections, remote usability tests have become more common. Here, the participant and the tester communicate via a collaboration tool, such as Microsoft PlaceWare. While remote tests can save travel costs, they can also reduce the quality of the data because there is no face-to-face interaction between the participant and the tester. However, as technology improves so will the quality of remote tests.

CONDUCTING A USABILITY TEST

What exactly is involved in a usability test? The following list shows a schematic of tasks that can be involved in most usability tests. The exact order and combination of tasks can differ from the order in which they’re presented here, but it’s a good idea to consider each of these tasks during the testing process.

- Obtain more than one copy of the product
- If no UI work has been done, perform an evaluation of the UI
- Define the scope of the test
- Identify the user population
- Set up the test platform
- Design the test methodology
- Define the tasks
- Determine usability measures
- Define usability criteria
- Define behavioral criteria
- Create questionnaires
- Create the test plan
- Present the test plan
- Recruit participants
- Run one or more pilot tests
- Run the test
- Analyze the results
- Create the report
- Consult on design changes
- Repeat as necessary

OBJECTIVE MEASURES

Objective measures are used to give an unbiased look at participant performance. Many measures are typically tracked during a test, although only two or three measures might be of use in the final analysis.

TIME-ON-TASK

Finding out the amount of time it takes to complete a task is a great way to gauge the progress of a design. As a product matures, tasks should be able to be performed faster in each new version. Recording time-on-task will help you determine whether this is true.

ERROR RATE/TYPE OF ERROR

Simply summing the number of errors will not provide a sensitive measure of usability. While some participants might be very methodical in their explorations, other participants might learn by trial-and-error only. So, individual differences among participants can lead to huge variations in errors.

Classifying the type of errors that are made can tell you much more about a product. Are errors due to something simple such as selecting an incorrect menu item? Or, are the errors due to something more complex such as the user not understanding the interaction model that is being presented?

REQUESTS FOR ASSISTANCE BY PARTICIPANTS

Participants usually request assistance after an error has occurred. In this instance, rather than having to infer what the issue might be, a participant tells you the nature of the issue.

ASSISTANCE PROVIDED BY THE TEST ADMINISTRATOR

There are times when some participants will veer off-course or become totally confused when testing. While leading participants to answers is not desirable, it is sometime necessary. A minor assist can be thought of as a hint to help participants move in the correct direction. For example, a test administrator might tell a participant to check the Help for the product or select a different branch of the menu. Given time, the participants probably would have found the answer by themselves.

Providing a major assist is just the opposite. Given time, the participants would not be able to complete the task on their own. The test administrator provides key information, without which the participants would not be able to complete the task.

If walk-through assistance is needed, the problem is serious. In this instance, even after major assistance has been provided, one or more participants still cannot complete a task. If a walk-through occurs on the same task for more than one participant, it indicates a serious issue with the product.

REFERENCES TO THE DOCUMENTATION

Many times, documentation is not available at the time a usability test is run, for example, for tests of prototypes. However, if documentation is available, it is always a good idea to track what information the participants are looking for and how they are using the information. In other words, is Help actually useful? Did the participants find what they were looking for? Are there terms missing from the index?

Participants can also be questioned as to what type of information they expect to find and where they expect to find it. With the prevalence of the Web, does the user even want to have hard-copy documentation, or would they prefer to have everything online?

WEB-LOGS/LINK ANALYSES/KEYSTROKE ANALYSES

These types of measures can tell you how an individual is using your product but are, by far, some of the most difficult measures to analyze. These objective measures can quickly generate large amounts of data, so their use depends on finding an expeditious way to classify the information.

One major issue with these types of data is that they tell you what users did but don't tell you why they did it. Analysis of the data can easily be influenced by the researcher's bias.

SUBJECTIVE MEASURES

To get the whole picture about product usability, it is imperative to balance objective measures with subjective measures.

IMPORTANCE/SATISFACTION

Typically, satisfaction is gathered via ratings scales. There are many different methods of determining satisfaction with a product, but the most popular method is to use a 5- or 7-point Likert scale.

An issue to take into consideration when using ratings scales is that users will often inflate their scores. Rather than rate their actual experience, they will rate their sense of accomplishment at mastering a difficult system.

OPEN-ENDED QUESTIONS

Ratings scales can be balanced with open-ended questionnaires. General questions such as "What did you like about the interface?" or "What did you dislike about the interface?" tend to elicit issues that are the most important to the participant.

You can also ask questions that are detailed and specific to parts of the interface or the tasks. This, of course, will give you more focused information. The trade-off is that detailed questionnaires can take a lot of time to complete.

COMMENTS FROM PARTICIPANTS

Comments made by participants during a test are a great source of information. The "think-aloud" protocol, where participants are encouraged to talk as they work, can generate a running commentary of the participants' thought

processes as they solve tasks. However, individual differences again play a factor, in that some individuals are not comfortable talking while they work.

THE SAS USABILITY LAB

The SAS Usability Laboratory opened in the fall of the year 2000. Since then, between 10 and 20 usability evaluations have been run each year, and hundreds of individuals participated in the tests. Participants are observed through a one-way mirror and by using video cameras. A scan converter is attached to the participants' PCs to enable the PC video signal to be converted to a television signal for recording.

In a typical usability test, a test administrator and a product representative are in the Control Room and the participant is in a Test Cell. Figure 1 shows the SAS Usability Laboratory Test Cell and Control Rooms. The Observation Room (not shown) is set up like a conference room, with an LCD projection screen that shows the image on the participants' PCs and a television that shows the image from the camera.

Test Cell

Control Room

Figure 1. The SAS Usability Laboratory

ACTIVEX GRAPH MENU EVALUATION

This evaluation was comprised of three separate tests. First, a baseline test of the original design of the ActiveX Graph Control was completed. The original design used multiple context menus, which were displayed by right-clicking an area of the graph. A different context menu displayed based on where the pointer was hovering on the graph. The primary problem with this design was that it placed a memory burden on the user. The participants were able to complete the tasks, but it took multiple tries to find the pop-up menu that contained the correct menu choice.

The second design of the ActiveX Graph Control integrated the various context menus into a single, tabbed context menu. This design operated much better than the first design. The speed with which participants could move between menus greatly reduced the memory load. Overall performance was significantly improved.

However, it was discovered that performance still suffered on a task that asked participants to change the labels on a pie chart. Participants were still completing this task by trial-and-error.

Analysis of the data showed that the problem was due to a single label. The word "Slice" labeled a drop-down list that was used to change the value that was displayed in a pie-chart segment. While all of the participants eventually completed the task successfully, none of them reported being familiar with the term "Slice" in this context.

The third design was a duplicate of the tabbed context menu design, with the exception that the "Slice" label was replaced with "Percent". The participants' familiarity with the term enabled them to complete the task with little or no learning. The improvement in performance was dramatic. What initially took minutes to master was now completed in seconds.

SAS DISTRIBUTED CLIENT APPLICATION

This effort is an example of the role of usability testing in a truly iterative design process. For the first version of this product, a usability test was conducted once a week for a two-month period. Because this was a new product, the tests focused on various components rather than the whole system. This enabled development to start work on one feature while another feature was being designed.

Each Thursday, four or five people participated in tests. The data was analyzed on Friday morning and a meeting convened on Friday afternoon to discuss the results and the direction for the next week. Programming changes were made between Monday and Wednesday, and on the following Thursday, the next version was tested.

Rather than relying on a single design and hope that it had the best interaction model, this mixture of rapid prototyping and usability testing allowed a number of concepts to be evaluated. While some ideas worked and others did not, the information that was gathered during these tests can now be applied to future designs.

SAS 9.1.2 INSTALLATION KIT

This test was conducted by Ryan West and Sharon Stanners, User Interface Analysts who work in SAS R&D. At issue was the size of the SAS 9.1.2 Installation Kit, which had grown to include numerous pieces of documentation and CDs, and it had become unwieldy.

West and Stanners first conducted a baseline test with the current SAS Installation Kit. The 10 participants in this test were all SAS users. The goal of the participants was simple—to get SAS up-and-running.

In this test, it was found that users rarely read much, if any, of the documentation. The participants quickly scanned the documentation in the kit, found the Setup CD in the back, and inserted it in the CD drive. They then followed the on-screen setup instructions to complete the installation.

A second test looked at a new concept—a Quick Install Guide. This was a single sheet of paper, printed on front and back, that contained numbered installation steps. The participants were presented with the same task as in the first test. However, this time both the Installation Kit and the new Quick Install Guide were available to the participants. Of the eight people who were participating in the test, seven participants used the Quick Install Guide. Interestingly, steps that were overlooked in the first test (such as turning off virus protection before loading any programs) were performed this time.

CONCLUSION

Usability and the SAS Usability Laboratory have had a positive impact on SAS products. By involving SAS customers and representative users, we more thoroughly understand the needs and expectations of SAS users. Through early and iterative testing, SAS products are now easier to use.

The eventual goal of usability testing at SAS is to enable users to open any SAS product and find a familiar interface. When users already know how to perform tasks in the interface, the speed with which productive work can begin will be greatly improved, and users' personal performances will also improve.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author:

Paul Hankey
SAS Institute Inc.
SAS Campus Drive
Cary, NC 27513
Work Phone: 919-677-8000
paul.hankey@sas.com