CHAPTER

Special Topics

9

This chapter introduces a number of topics related to the measurement or analysis of usability data but not traditionally thought of as part of ''mainstream'' usability data. These include information you can glean from live data on a production website, data from card-sorting studies, data related to the accessibility of a website, the topic of Six Sigma and how it relates to usability, and usability Return on Investment (ROI). Our primary goal in this chapter is to make you aware of these topics, provide an overview of each of them, and then point you to additional resources for more detailed information.

9.1 LIVE WEBSITE DATA

If you're dealing with a live website, there's a potential treasure trove of data available to you about what the visitors to your site are actually doing—what pages they're visiting, what links they're clicking on, and what paths they're following through the site. The challenge usually isn't getting the raw data but making sense of it.

Entire books have been written only on the subject of web metrics and web analytics (e.g., Kaushik, 2007; Peterson, 2004; Sterne, 2002). There's even a *For Dummies* book on the topic (Sostre & LeClaire, 2007). So obviously we won't be able to do justice to the topic in just one section of one chapter in this book. What we'll try to do is introduce you to some of the things you can learn from live website data and specifically some of the implications they might have for the usability of your site.

9.1.1 Server Logs

Some websites get huge numbers of visitors every day. But regardless of how many visitors your site gets (assuming it gets some), you can learn from what they're doing on the site.

Early attempts at analyzing server log files focused on the number of requests to the server associated with each page. But each inline element of a page (e.g., each image) generates a separate request to the server, so these numbers can be misleading. Two metrics that generate more accurate numbers are *page views* and *visits* to a page. A *page view* is a request to the server for a specific page but not for its various components. A *visit* is a request for a specific page by a given user (commonly identified by IP address) within a specified time period (often 30 minutes).

Simply looking at the number of page views or visits for various pages in your site can be enlightening, especially over time or across iterations of the site. For example, assume that a page about Product A on your site was averaging 100 page views per day for a given month. Then you modified the homepage for your site, including the description of the link to Product A's page. Over the next month, the Product A page averaged 150 page views per day. It would certainly appear that the changes to the homepage significantly increased the number of visitors accessing the Product A page. But you need to be careful that other factors didn't cause the increase. For example, in the financial-services world, certain pages have seasonal differences in their number of page views. A page about contributions to an Individual Retirement Account (IRA), for example, tends to get more visits in the days leading up to April 15 because of the deadline for contributing to the prior year's IRA.

It's also possible that something caused your site as a whole to start getting more visitors, which certainly could be a good thing. But it could also be due to factors not related to the design or usability of your site, such as news events related to the subject matter of your site. This also brings up the issue of the impact that search "bots" can have on your site's statistics. Search bots, or spiders, are automated programs used by most of the major search engines to "crawl" the web by following links and indexing the pages they access. One of the challenges, once your site becomes popular and is being "found" by most of the major search engines, is filtering out the page visits due to these search bots. Most bots (e.g., Google, Yahoo!) usually identify themselves when making page requests and thus can be filtered out of the data.

What analyses can be used to determine if one set of page views is significantly different from another set? Consider the data shown in Table 9.1, which shows the number of page views per day for a given page over two different weeks. Week 1 was before a new homepage with a different link to the page in question was launched, and Week 2 was after. The new homepage contained different wording for the link to this page.

These data show a pattern that's typical in some web statistics, which is a difference in the number of page views for the weekend versus the weekdays. These data can be analyzed using a paired *t*-test to see if the average for Week 2 (519 page views per day) is significantly different from the average for Week 1 (454 page views per day). It's important to use a paired *t*-test because of the variability due to the days of the week; comparing each day to itself from the previous week takes out the variability due to days. A paired *t*-test shows that this difference is statistically significant ($p < .01$). (See section 2.5.2 for details on how to run a paired *t*-test in Excel.)

Table 9.1 Numbers of Page Views for a Given Web Page over Two Different Weeks

	Week 1	Week 2
Sunday	237	282
Monday	576	623
Tuesday	490	598
Wednesday	523	612
Thursday	562	630
Friday	502	580
Saturday	290	311

9.1.2 Click-Through Rates

Click-through rates can be used to measure the effectiveness of different ways of presenting a link. They indicate the percentage of visitors who are shown a particular link who then actually click on it. If a link is shown 100 times and it is clicked on one of those times, its click-through rate is 1 percent. Most commonly the term is used to measure the effectiveness of web ads, but the concept applies to any link. For example, Nielsen (2001b) describes several different approaches his company took to promoting their usability conference in London. Figure 9.1 shows two different versions of search engine ads (triggered when a user entered appropriate keywords) that they tried.

The click-through rate for the ad that was more specific ("Jakob Nielsen in Europe," dates included) was 55 percent greater than the more general one. Given the number of visitors many search engines get, this could be a huge difference in the actual number of users who clicked on the ad.

What analyses can be used to determine if the click-through rate for one link is significantly different from that for another link? One such analysis is the chi-square

FIGURE 9.1

Two different search engine ads tested for promoting the Nielsen Norman Group's London usability conference. The percentages indicate the click-through rates for the European users. *Source:* From Nielsen (2001b); used with permission.

Table 9.2 Click-Through Rates for Two Different Links

	Click	No Click
Link 1	145	10,289
Link 2	198	11,170

Table 9.3 Same Data as in Table 9.2 with Row and Column Sums Added

Observed	Click	No Click	Sum
Link 1	145	10,289	**10,434**
Link 2	198	11,170	**11,368**
Sum	**343**	**21,459**	**21,802**

test. A chi-square test lets you determine whether an observed set of frequencies is significantly different from an expected set of frequencies (see section 2.7.1 for more details). For example, consider the data shown in Table 9.2 that represent click rates for two different links: the number of times each link was clicked and the number of times each was presented but not clicked. The click-through rate for Link 1 is 1.4 percent [145 / (145 + 10,289)]. The click-through rate for Link 2 is 1.7 percent [198 / (198 + 11,170)]. But are these two significantly different from each other? Link 2 got more clicks, but it was also presented more times.

To do a chi-square test, you must first construct a table of expected frequencies as if there were no difference in the click-through rates of Link 1 and Link 2. This is done using the sums of the rows and columns of the original table, as shown in Table 9.3. The data are used to calculate the expected frequencies if there was no difference in the click-through rates.

By taking the product of each pair of row and column sums and dividing that by the grand total, you get the expected values as shown in Table 9.4. For example, the expected frequency for a "Click" on "Link 1" (164.2) is the product of the respective row and column sums divided by the grand total: $(343 \times 10{,}434)/21{,}802$. The "CHITEST" function in Excel can then be used to compare the actual frequencies in Table 9.2 to the expected frequencies in Table 9.4. The resulting value is $p = 0.037$, indicating that there is a significant difference between the click-through rates for Link 1 and Link 2.

You should keep two important points about the chi-square test in mind. First, the chi-square test must be done using raw frequencies or counts, *not* percentages. You commonly think of click-through rates in terms of percentages, but that's not how you test for significant differences between them. Also, the categories used must be mutually exclusive and exhaustive. That's why the preceding example

Table 9.4 Expected Frequencies with No Difference in Click-Through Rates for Link 1 and Link 2

Expected	Click	No Click
Link 1	164.2	10,269.8
Link 2	178.8	11,189.2

Note: Data shown are derived from the sums shown in Table 9.3.

used "Click" and "No Click" as the two categories of observations for each link. Those two categories are mutually exclusive, and they account for all possible actions that could be taken on the link: either the user clicked on it or didn't.

9.1.3 Drop-Off Rates

Drop-off rates can be a particularly useful way of detecting where there might be some usability problems on your site. The most common use of drop-off rates is to identify where in a sequence of pages users are dropping out of or abandoning a process, such as opening an account or completing a purchase. For example, assume that the user must fill out the information on a sequence of five pages to open some type of account. Table 9.5 reflects the percentage of users who started the process who actually completed each of the five pages.

In this example, all of the percentages are relative to the number of users who started the entire process—that is, who got to Page 1. So 89 percent of the users who got to Page 1 successfully completed it, 80 percent of that original number completed Page 2, and so on. Given the data in Table 9.5, which of the five pages do the users seem to be having the most trouble with? The key is to look at how many users dropped off from each page—in other words, the difference between how many got to the page and how many completed it. Those "drop-off percentages" for each of the pages are shown in Table 9.6.

Table 9.5 Percentage of Users Who Started and Completed Each Step in a Multipage Process

Page 1	89%
Page 2	80%
Page 3	73%
Page 4	52%
Page 5	49%

Table 9.6 Drop-Off Percentages for Each Page Shown in Table 9.5	
Page 1	11%
Page 2	9%
Page 3	7%
Page 4	21%
Page 5	3%

This makes it clear that the largest drop-off rate, 21 percent, is associated with Page 4. If you're going to redesign this multipage process, you would be well advised to learn what's causing the drop-off at Page 4 and then try to address that in the redesign.

9.1.4 A/B Studies

A/B studies are a special type of live-site study in which you manipulate the pages that users see. The traditional approach to A/B testing on a website involves posting two alternative designs for a given page. Some visitors to the site see the "A" version and others see the "B" version. In many cases, this assignment is random, so about the same number of visitors sees each version. In some cases, the majority of visitors see the existing page, and a smaller percentage see an experimental version that's being tested. Although these studies are typically called A/B tests, the same concept applies to any number of alternative designs for a page.

Technically, visitors to a page can be directed to one of the alternative pages in a variety of ways, including on random number generation, the exact time (e.g., an even or odd number of seconds since midnight), or several other techniques. Typically, a cookie is set to indicate which version the visitor was shown so that if he or she returns to the site within a specified time period, the same version will be shown again. Keep in mind that it's important to test the alternative versions at the same time because of the external factors mentioned before that could affect the results if you tested at different times.

Usborne (2005) described an A/B/C test of three different versions of a page explaining a report and including a form to complete for purchasing the report. Page A was the existing page. Page B used the same basic page layout and design but modified the text or "copy." Page C was a redesign that involved changing from a single-column layout to a two-column layout; the goal was to bring more of the key content "above the fold." These three versions of the same basic page were then tested simultaneously. The key metric for comparing the three versions was the number of sales of the report each generated. The results are shown in Table 9.7

Table 9.7 Data from an A/B/C Test

	Page A	Page B	Page C
Percent of traffic	34%	33%	33%
New sales	244	282	114
Change relative to A	N/A	15.6%	53.3%

Source: Adapted from Usborne (2005); used with permission.

The "Percent of traffic" reflects the fact that each of the three versions of the page was shown the same number of times. The key is the number of "New sales," which is the number of visitors who actually purchased the report. Since Page A was the existing page, the others were compared to it: Page B resulted in a 15.6 percent increase in sales, whereas Page C resulted in a 53.3 percent *decrease* in sales! The lesson to learn from this is that the actual text on a page really makes a difference (as shown by the improvement with Page B) and that sometimes the design and layout decisions you make may not always have the effect you intend (as in Page C).

Carefully designed A/B tests can give you significant insight into what works and what doesn't work on your website. Many companies are constantly doing A/B tests on their live sites, although most users don't notice it. In fact, as Kohavi and Round (2004) explained, A/B testing is constant at Amazon, and experimentation through A/B testing is one of the ways they make changes to their site.

9.2 CARD-SORTING DATA

Card-sorting as a technique for organizing the elements of an information system in a way that makes sense to the users has been around at least since the early 1980s. For example, Tullis (1985) used the technique to organize the menus of a mainframe operating system. More recently, the technique has become popular as a way of informing decisions about the information architecture of a website (e.g., Maurer & Warfel, 2004). Over the years the technique has evolved from being a true card-sorting exercise using index cards to an online exercise using virtual cards. Although many usability professionals are familiar with the basic card-sorting techniques, fewer seem to be aware that various metrics can be used in the analyses of card-sorting data.

The two major types of card-sorting exercises are (1) open card-sorts, where you give participants the cards that are to be sorted but let them define their own groups that the cards will be sorted into, and (2) closed card-sorts, where you give participants the cards to be sorted as well as the names of the groups to sort them into. Although some metrics apply to both, others are unique to each.

CARD-SORTING TOOLS

A number of tools are available for conducting card-sorting exercises. Some are desktop applications and others are web-based. Here are some of the ones we're familiar with:

- CardSort (*http://www.cardsort.net/*)—a Windows application
- CardZort (*http://www.cardzort.com/cardzort/*)—a Windows application
- Classified (*http://www.infodesign.com.au/usabilityresources/classified/*)—a Windows application
- OptimalSort (*http://www.optimalsort.com/*)—a web-based service
- UzCardSort (*http://uzilla.mozdev.org/cardsort.html*)—a Mozilla extension
- WebCAT (*http://zing.ncsl.nist.gov/WebTools/WebCAT/overview.html*)—a free web-based tool you can install on your server if you're a techno-geek!
- Websort (*http://www.websort.net/*)—a web-based service
- XSort (*http://www.ipragma.com/xsort*)—a Mac OS X application

Although not a card-sorting tool, you can also use PowerPoint or similar programs to do card-sorting exercises. Simply create a slide that has the cards to be sorted along with empty boxes and then e-mail that to participants, asking them to put the cards into the boxes and to name the boxes. Then they simply e-mail the file back.

9.2.1 Analyses of Open Card-Sort Data

One way to analyze the data from an open card-sort is to create a matrix of the "perceived distances" among all pairs of cards in the study. For example, assume you conducted a card-sorting study using ten fruits: apples, oranges, strawberries, bananas, peaches, plums, tomatoes, pears, grapes, and cherries. Assume one participant in the study created the following groups:

- "Large, round fruits": apples, oranges, peaches, tomatoes
- "Small fruits": strawberries, grapes, cherries, plums
- "Funny-shaped fruits": bananas, pears

You can then create a matrix of "perceived distances" among all pairs of the fruits for each participant by using the following rules:

- If this person put a pair of cards in the same group, it gets a distance of 0.
- If this person put a pair of cards into different groups, it gets a distance of 1.

Using these rules, the distance matrix for the preceding participant would look like what's shown in Table 9.8.

We're only showing the top half of the matrix for simplicity, but the bottom half would be exactly the same. The diagonal entries are not meaningful because the distance of a card from itself is undefined. (Or it can be assumed to be zero if needed in the analyses.) So for any one participant in the study, the entries in this matrix will only be 0's or 1's. The key is to then combine these matrices for all the

Table 9.8 Distance Matrix for a Participant's Card-Sorting Results

	Apples	Oranges	Strawberries	Bananas	Peaches	Plums	Tomatoes	Pears	Grapes	Cherries
Apples	—	0	1	1	0	1	0	1	1	1
Oranges		—	1	1	0	1	0	1	1	1
Strawberries			—	1	1	0	1	1	0	0
Bananas				—	1	1	1	0	1	1
Peaches					—	1	0	1	1	1
Plums						—	1	1	0	0
Tomatoes							—	1	1	1
Pears								—	1	1
Grapes									—	0
Cherries										—

participants in the study. Let's assume you had 20 participants do the card-sorting exercise with the fruits. You can then sum the matrices for the 20 participants. This will create an overall distance matrix whose values can, in theory, range from 0 (if all participants put that pair into the same group) to 20 (if all participants put that pair into different groups).

Table 9.9 shows an example of what that might look like. In this example, only 2 of the participants put the oranges and peaches in different groups, whereas all 20 of the participants put the bananas and tomatoes into different groups.

A CARD-SORT ANALYSIS SPREADSHEET

Donna Maurer has developed an Excel spreadsheet for the analysis of card-sorting data. She uses some very different techniques for exploring the results of a card-sorting exercise than the more statistical techniques we're describing here, including support for the person doing the analysis to standardize the categories by grouping the ones that are similar. The spreadsheet and instructions can be downloaded from *http://www.rosenfeldmedia.com/books/cardsorting/blog/card_sort_analysis_spreadsheet/*.

This overall matrix can then be analyzed using any of several standard statistical methods for studying distance (or similarity) matrices. Two that we find useful are hierarchical cluster analysis (e.g., Aldenderfer & Blashfield, 1984) and multidimensional scaling, or MDS (e.g., Kruskal & Wish, 2006). Both are available in a variety of commercial statistical analysis packages. One that we use is NCSS (NCSS, 2007), which was used for both of the following analyses.

Hierarchical Cluster Analysis

The goal of hierarchical cluster analysis is to build a tree diagram where the cards that were viewed as most similar by the participants in the study are placed on branches that are close together. For example, Figure 9.2 shows the result of a hierarchical cluster analysis of the data in Table 9.9. The key to interpreting a hierarchical cluster analysis is to look at the point at which any given pair of cards "join together" in the tree diagram. Cards that join together sooner are more similar to each other than those that join together later. For example, the pair of fruits with the lowest distance in Table 9.9 (peaches and oranges; distance = 2) join together first in the tree diagram.

Something to be aware of is that several different algorithms can be used in hierarchical cluster analysis to determine how the "linkages" are created. Most of the commercial packages that support hierarchical cluster analysis let you choose which method to use. The linkage method we think works best is one called the Group Average method. But you might want to experiment with some of the other linkage methods to see what the results look like; there's no absolute rule saying one is better than another.

One thing that makes hierarchical cluster analysis so appealing for use in the analysis of card-sorting data is that you can use it to directly inform how you might

Table 9.9 Overall Distance Matrix for 20 Fruit Card-Sorting Study Participants

	Apples	Oranges	Strawberries	Bananas	Peaches	Plums	Tomatoes	Pears	Grapes	Cherries
Apples	—	5	11	16	4	10	12	8	11	10
Oranges		—	17	14	2	12	15	11	12	14
Strawberries			—	17	16	8	18	15	4	8
Bananas				—	17	15	20	11	14	16
Peaches					—	9	11	6	15	13
Plums						—	12	10	9	7
Tomatoes							—	16	18	14
Pears								—	12	14
Grapes									—	3
Cherries										—

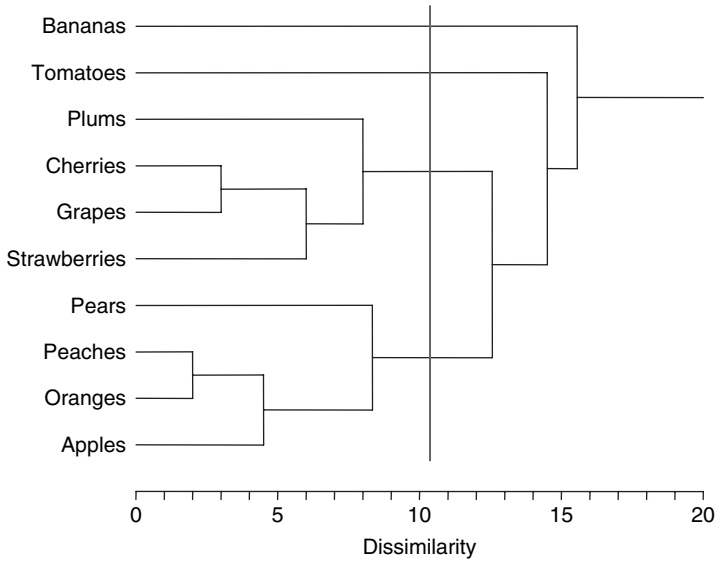


FIGURE 9.2

Result of a hierarchical cluster analysis of the data shown in Table 9.9.

organize the cards (pages) in a website. One way to do this is to take a vertical "slice" through the tree diagram and see what groupings that creates. For example, Figure 9.2 shows a 4-cluster "slice": The vertical line intersects four horizontal lines, forming the four groups—(1) bananas; (2) tomatoes; (3) plums, cherries, grapes, and strawberries; and (4) pears, peaches, oranges, and apples. How do you decide how many clusters to create when taking a "slice" like this? Again, there's no fixed rule, but one method we like is to calculate the average number of groups of cards created by the participants in the card-sorting study and then try to approximate that.

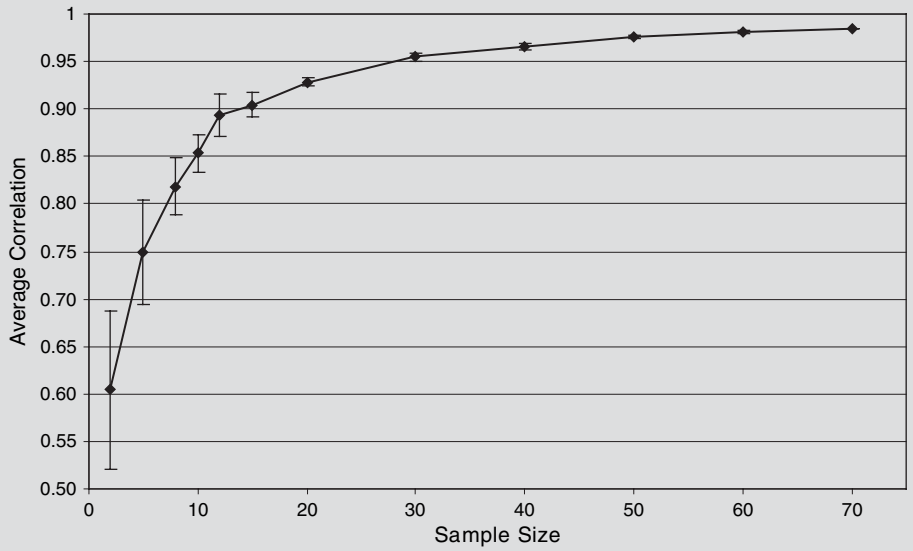
After taking a "slice" through the tree diagram and identifying the groups created by that, the next thing you might want to do is determine how those groups compare to the original card-sorting data—in essence, to come up with a "goodness of fit" metric for your derived groups. One way of doing that is to compare the pairings of cards in your derived groups with the pairings created by each participant in the card-sorting study and to identify what percentage of the pairs match. For example, for the data in Table 9.8, only 7 of the 45 pairs do *not* match those identified in Figure 9.2. The 7 nonmatching pairings are apples–tomatoes, apples–pears, oranges–tomatoes, oranges–pears, bananas–pears, peaches–tomatoes, and peaches–pears. That means 38 pairings do match, or 84 percent (38/45). Averaging these matching percentages across all the participants will give you a measure of the goodness of fit for your derived groups relative to the original data.

Multidimensional Scaling

Another way of analyzing and visualizing the data from a card-sorting exercise is using multidimensional scaling, or MDS. Perhaps the best way to understand MDS

HOW MANY PARTICIPANTS ARE ENOUGH FOR A CARD-SORTING STUDY?

Tullis and Wood (2004) conducted a card-sorting study in which they addressed the question of how many people you need for a card-sorting study if you want to get reliable results from your analyses. They did an open sort with 46 cards and 168 participants. They then analyzed the results for the full dataset (168 participants) as well as many random subsamples of the data from 2 to 70 participants. The correlations of the results for those subsamples to the full dataset looked like the chart here.



The "elbow" of that curve appears to be somewhere between 10 and 20, with a sample size of 15 yielding a correlation of 0.90 with the full dataset. Although it's hard to know how well these results would generalize to other card-sorting studies with different subject matter or different numbers of cards, they at least suggest that about 15 may be a good target number of participants.

is through an analogy. Imagine that you had a table of the mileages between all pairs of major U.S. cities but not a map of where those cities are located. An MDS analysis could take that table of mileages and derive an approximation of the map showing where those cities are relative to each other. In essence, MDS tries to create a map in which the distances between all pairs of items match the distances in the original distance matrix as closely as possible.

The input to an MDS analysis is the same as the input to a hierarchical cluster analysis—a distance matrix, like the example shown in Table 9.9. The result of an MDS analysis of the data in Table 9.9 is shown in Figure 9.3. The first thing that's apparent from this MDS analysis is how the tomatoes and bananas are isolated from

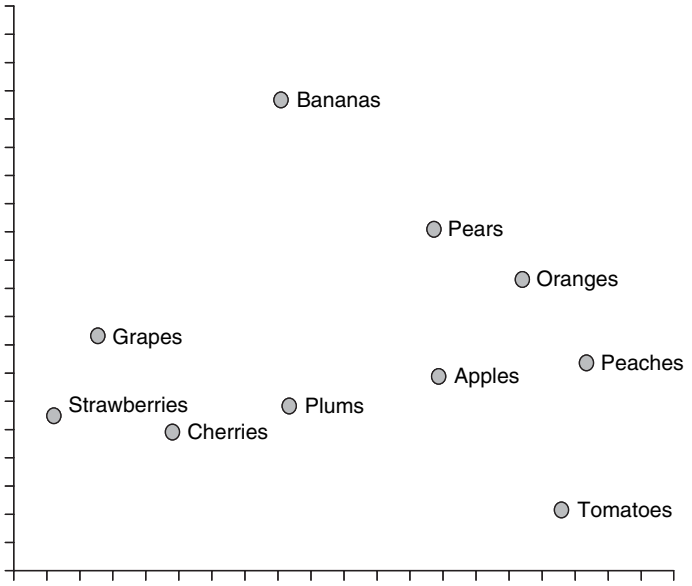


FIGURE 9.3

MDS analysis of the distance matrix in Table 9.9.

all the other fruit. That's consistent with the hierarchical cluster analysis, where those two fruits were the last two to join all the others. In fact, our 4-cluster "slice" of the hierarchical cluster analysis (Figure 9.2) had these two fruits as groups unto themselves. Another thing apparent from the MDS analysis is how the strawberries, grapes, cherries, and plums cluster together on the left, and the apples, peaches, pears, and oranges cluster together on the right. That's also consistent with the hierarchical cluster analysis.

Notice that it's also possible to use more than two dimensions in an MDS analysis, but we've rarely seen a case where adding even just one more dimension yields particularly useful insights into card-sorting data. Another point to keep in mind is that the orientation of the axes in an MDS plot is arbitrary. You could rotate or flip the map any way you want, and the results would still be the same. The only thing that's actually important is the relative distances between all pairs of the items.

The most common metric that's used to represent how well an MDS plot reflects the original data is a measure of "stress" that's sometimes referred to as *Phi*. Most of the commercial packages that do MDS analysis can also report the stress value associated with a solution. Basically, it's calculated by looking at all pairs of items, finding the difference between each pair's distance in the MDS map and its distance in the original matrix, squaring that difference, and summing those squares. That measure of stress for the MDS map shown in Figure 9.3 is 0.04. The smaller the value, the better. But how small does it really need to be? A good rule of

thumb is that stress values under 0.10 are excellent, whereas stress values above 0.20 are poor.

We find that it's useful to do both a hierarchical cluster analysis and an MDS analysis. Sometimes you see interesting things in one that aren't apparent in the other. And they are different statistical analysis techniques, so you shouldn't expect them to give exactly the same answers. For example, one thing that's sometimes easier to see in an MDS map is which cards are "outliers"—those that don't obviously belong with a single group. There are at least two reasons why a card could be an outlier: (1) It could truly be an outlier—a function that really is different from all the others, or (2) it could have been "pulled" toward two or more groups. When designing a website, you would probably want to make these functions available from *each* of those areas.

9.2.2 Analyses of Closed Card-Sort Data

Closed card-sorts, where you give participants not only the cards but also the names of the groups in which to sort them, are probably done less often than open card-sorts. Typically, you would start with an open sort to get an idea of the kinds of groups that users would naturally create and the names they might use for them. Sometimes it's helpful to follow up an open sort with one or more closed sorts, mainly as a way of testing your ideas about organizing the functions. With a closed card-sort you have an idea about how you want to organize the functions, and you want to see how close users come to matching the organization you have in mind.

We recently used closed card-sorting to compare different ways of organizing the functions for a website (Tullis, 2007). We first conducted an open sort with 54 functions. We then used those results to generate six different ways of organizing the functions that we then tested in six parallel closed card-sorting exercises. Each closed card-sort used the same 54 functions but presented different groups to sort the functions into. The number of groups in each "framework" (set of group names) ranged from three to nine. Each participant only saw and used one of the six frameworks.

In looking at the data from a closed card-sort, the main thing you're interested in is how well the groups "pulled" the cards to them that you intend to belong to those groups. For example, consider the data in Table 9.10, which shows the percentage of the participants in a closed card-sorting exercise who put each of the ten cards into each of the three groups provided.

The other percentage, shown on the right in Table 9.10, is the highest percentage for each card. This is an indicator of how well the "winning" group pulled the appropriate cards to it. What you hope to see are cases like Card 10 in this table, which was very strongly pulled to Group C, with 92 percent of the participants putting it in that group. The ones that are more troubling are cases like Card 7, where 46 percent of the participants put it in Group A, but 37 percent put it in Group C—so participants were very "split" in deciding where that card belonged in this set of groups.

Table 9.10 Percentage of Closed Card-Sort Participants Who Put Each Card in Each of the Three Groups Provided

Card	Group A	Group B	Group C	Max
1	17%	78%	5%	**78%**
2	15%	77%	8%	**77%**
3	20%	79%	1%	**79%**
4	48%	40%	12%	**48%**
5	11%	8%	81%	**81%**
6	1%	3%	96%	**96%**
7	46%	16%	37%	**46%**
8	57%	38%	5%	**57%**
9	20%	75%	5%	**75%**
10	4%	5%	92%	**92%**
			Average	**73%**

One metric you could use for characterizing how well a particular set of group names fared in a closed card-sort is the average of these maximum values for all the cards. For the data in Table 9.10, that would be 73 percent. But what if you want to compare the results from closed card-sorts with the same cards but different sets of groups? That average maximum percentage will work well for comparisons as long as each set contained the same number of groups. But if one set had only three groups and another had nine groups, as in the Tullis (2007) study, it's not a fair metric for comparison. If participants were simply acting randomly in doing the sorting with only three groups, by chance they would get a maximum percentage of 33 percent. But if they were acting randomly in doing a sort with nine groups, they would get a maximum percentage of only 11 percent. So using this metric, a framework with more groups is at a disadvantage in comparison to one with fewer groups.

We experimented with a variety of methods to correct for the number of groups in a closed card-sort. The one that seems to work best is illustrated in Table 9.11. These are the same data as shown earlier in Table 9.10 but with two additional columns. The "2nd Place" column gives the percentage associated with the group that had the next-highest percentage. The "Difference" column is simply the difference between the maximum percentage and the 2nd-place percentage. A card that was pulled strongly to one group, such as Card 10, gets a relatively small penalty in this scheme. But a card that was more evenly split, such as Card 7, takes quite a hit.

Table 9.11 Same Data as in Table 9.10 with an Additional Two Columns

Card	Category A	Category B	Category C	Max	2nd Place	Difference
1	17%	78%	5%	**78%**	**17%**	**61%**
2	15%	77%	8%	**77%**	**15%**	**62%**
3	20%	79%	1%	**79%**	**20%**	**60%**
4	48%	40%	12%	**48%**	**40%**	**8%**
5	11%	8%	81%	**81%**	**11%**	**70%**
6	1%	3%	96%	**96%**	**3%**	**93%**
7	46%	16%	37%	**46%**	**37%**	**8%**
8	57%	38%	5%	**57%**	**38%**	**18%**
9	20%	75%	5%	**75%**	**20%**	**55%**
10	4%	5%	92%	**92%**	**5%**	**87%**
			Average	**73%**		**52%**

Note: The 2nd Place column refers to the next-highest percentage after the maximum percentage, and the Difference column indicates the difference between the maximum percentage and the 2nd-Place percentage.

The average of these differences can then be used to make comparisons between frameworks that have different numbers of groups. For example, Figure 9.4 shows the data from Tullis (2007) plotted using this method. We call this a measure of the percent agreement among the participants about which group each card belongs to. Obviously, higher values are better.

Notice that the data from a closed card-sort can also be analyzed using hierarchical cluster analysis and MDS analysis, just like the data from an open card-sort. These give you visual representations of how well the framework you presented to the participants in the closed card-sort actually worked for them.

9.3 ACCESSIBILITY DATA

Accessibility usually refers to how effectively someone with disabilities can use a particular system, application, or website (e.g., Henry, 2007; Kirkpatrick et al., 2006). We believe that accessibility is really just usability for a particular set of users. When viewed that way, it becomes obvious that most of the other metrics discussed in this book (e.g., task completion rates and times, self-reported metrics) can be applied to measure the usability of any system for users with different types

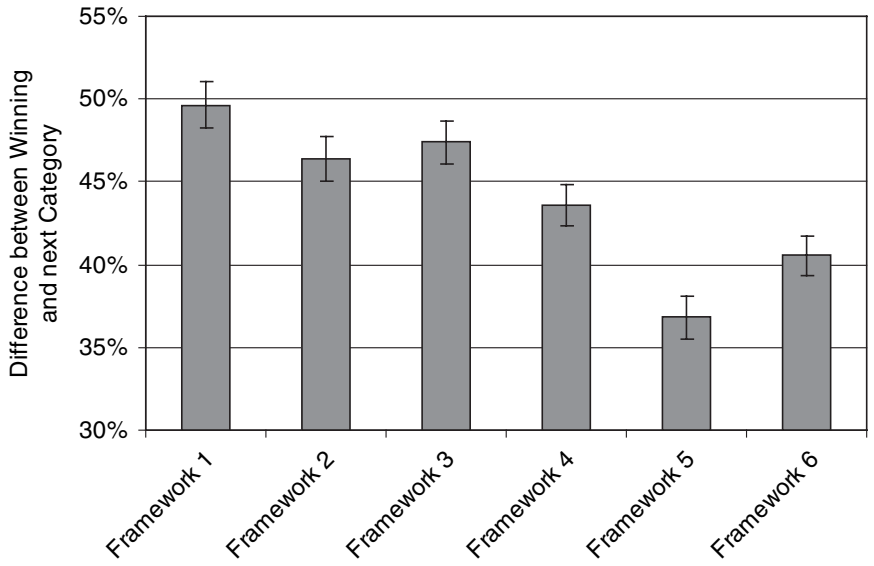


FIGURE 9.4

Comparison of six frameworks in six parallel closed card-sorts. Since the frameworks had different numbers of groups, a correction was used in which the percentage associated with the 2nd-place group was subtracted from the winning group. *Source:* Adapted from Tullis (2007); used with permission.

Table 9.12 Website Usability Test Results for Normal, Blind, and Low-Vision Users

	Screen Reader Users	Screen Magnifier Users	Control
Task Success	12.5%	21.4%	78.2%
Task Time	16:46	15:26	7:14
Errors	2	4.5	0.6
Subjective Rating	2.5	2.9	4.6

Source: Adapted from Nielsen (2001c); used with permission.

of disabilities. For example, Nielsen (2001c) reported four usability metrics from a study of 19 websites with three groups of users: blind users, who accessed the sites using screen-reading software; low-vision users, who accessed the sites using screen-magnifying software; and a control group who did not use assistive technology. Table 9.12 shows the results for the four metrics.

These results point out that the usability of these sites is far worse for the screen-reader and screen-magnifier users than it is for the control users. But the other important message is that the best way to measure the usability of a system or website for users with disabilities is to actually test with representative users. Although that's a very desirable objective, most designers and developers don't have the resources to test with representative users from all the disability groups that might want to use their product. That's where accessibility guidelines can be helpful.

Perhaps the most widely recognized web accessibility guidelines are the Web Content Accessibility Guidelines (WCAG) from the World-Wide Web Consortium (W3C) (World-Wide Web Consortium, 1999). These guidelines are divided into three categories:

Priority 1: Sixteen guidelines that ***must*** be met
Priority 2: Thirty guidelines that ***should*** be met
Priority 3: Nineteen guidelines that ***may*** be met

One way of quantifying how well a website meets these criteria is to assess how many of the pages in the site fail one of more of each of these guidelines.

Some automated tools can check for certain obvious violations of these guidelines (e.g., missing "Alt" text on image). Although the errors they detect are generally true errors, they also commonly miss many errors. Many of the items that the automated tools flag as ***warnings*** may in fact be true errors, but it takes a human to find out. For example, if an image on a web page has null Alt text defined (ALT=""), that may be an error if the image is informational, or it may be correct if the image is purely decorative. The bottom line is that the only really accurate way to determine whether accessibility guidelines have been met is by manual inspection of the code or by evaluation using a screen reader or other appropriate assistive technology. Often both techniques are needed.

AUTOMATED ACCESSIBILITY-CHECKING TOOLS

Some of the tools available for checking web pages for accessibility errors include the following:

- Bobby (*http://www.watchfire.com/products/webxm/bobby.aspx*); there's also a free 1-page checking version available at *http://webxact.watchfire.com/*
- Cynthia Says (*http://www.contentquality.com/*)
- HiSoftware's AccMonitor (*http://www.hisoftware.com/products/access_101.htm*)
- Accessibility Valet Demonstrator (*http://valet.webthing.com/access/url.html*)
- WebAIM's WAVE tool (*http://www.wave.webaim.org/wave35/index.jsp*)
- University of Toronto Web Accessibility Checker (*http://checker.atrc.utoronto.ca/index.html*)
- TAW Web Accessibility Test (*http://www.tawdis.net/taw3/cms/en*)

Once you've analyzed the pages against the accessibility criteria, one way of summarizing the results is to count the number of pages with the different types of errors. In most cases, the number of pages containing errors is a more meaningful metric than the actual number of errors. For example, Figure 9.5 shows the results of an analysis of a website against all three priorities of the WCAG guidelines. This shows that only 21 percent of the pages have no errors, whereas 30 percent have the most severe (Priority 1) errors. Notice that any given page may contain multiple errors of different priority levels. Each page containing errors is categorized according to the most severe errors on the page. So, for example, a page containing any Priority 1 errors is categorized as Priority 1 regardless of any other errors on the page. A page categorized as Priority 3 cannot have any Priority 1 or 2 errors.

Another useful way of analyzing and reporting accessibility errors is by counting the total number of errors (of whatever type you are checking for) per page and then looking at the number of pages containing various numbers of errors, as shown in Figure 9.6. The logic behind this analysis is that pages with significantly more errors tend to represent more of a barrier to access than those with fewer errors.

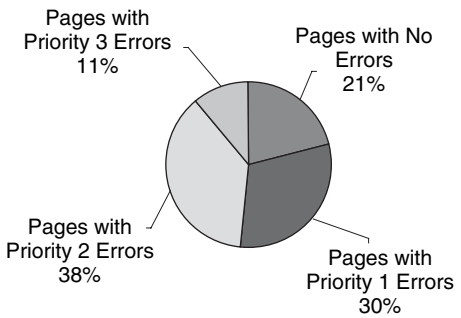


FIGURE 9.5

Results of analysis of a website against all three priority levels of the WCAG guidelines.

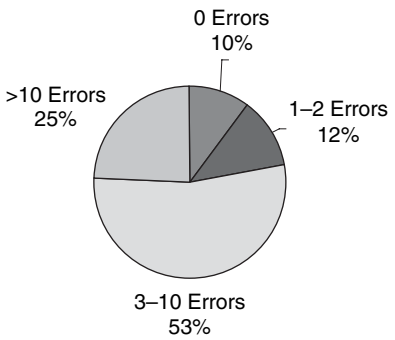


FIGURE 9.6

Data from an accessibility analysis showing the percentage of pages containing no errors, one to two errors, three to ten errors, and more than ten errors.

In the United States, another important set of accessibility guidelines is the so-called Section 508 guidelines, or, technically, the 1998 Amendment to Section 508 of the 1973 Rehabilitation Act (Section 508, 1998; also see Mueller, 2003). This law requires federal agencies to make their information accessible to people with disabilities, including what's on their websites. The law applies to all federal agencies when they develop, procure, maintain, or use electronic and information technology. Section 508 specifies 16 standards that websites must meet. Although there is significant overlap with the Priority 1 WCAG guidelines, Section 508 also includes some standards not listed in WCAG. As with the WCAG guidelines, we believe the most useful metric is a page-level metric, indicating whether the page passes all 16 standards or not. You can then chart the percentage of pages that pass versus those that fail.

9.4 RETURN-ON-INVESTMENT DATA

A book about usability metrics wouldn't be complete without at least some discussion of Return on Investment (ROI), since the usability metrics discussed in this book often play a key role in calculating ROI. But because entire books have been written on this topic (Bias & Mayhew, 2005; Mayhew & Bias, 1994), our purpose is to just introduce some of the concepts.

The basic idea behind usability ROI, of course, is to calculate the financial benefit attributable to usability enhancements for a product, system, or website. These benefits are usually derived from such measures as increased sales, increased productivity, or decreased support costs that can be attributed to the usability improvements. The key is to identify the cost associated with the usability improvements and then compare those to the financial benefits.

To illustrate some of the issues and techniques in calculating usability ROI, we'll look at an example from Diamond Bullet Design (Withrow, Brinck, & Speredelozzi, 2000). This case study involved the redesign of a state government web portal. The researchers conducted usability tests of the original website and a new version that had been created using a user-centered design process. The same ten tasks were used to test both versions. A few of them were as follows:

- You are interested in renewing a [*State*] driver's license online.
- How do nurses get licensed in [*State*]?
- To assist in traveling, you want to find a map of [*State*] highways.
- What four-year colleges are located in [*State*]?
- What is the state bird of [*State*]?

Twenty residents of the state participated in the study, which was a between-subjects design (with half using the original site and half using the new). The data collected included task times, task completion rates, and various self-reported metrics. The researchers found that the task times were significantly shorter for

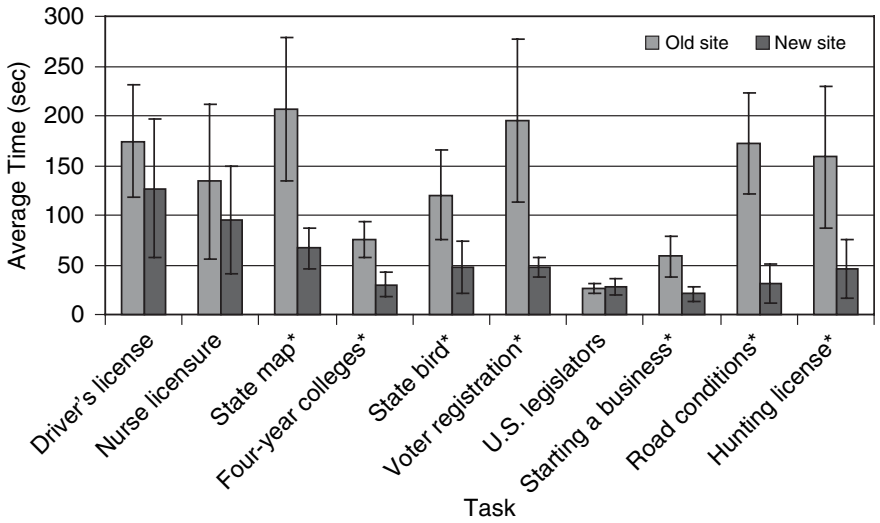


FIGURE 9.7

Task times for the original and the redesigned sites (* = Significant difference).
Source: Adapted from Withrow et al. (2000); used with permission.

Table 9.13 Summary of the Task Performance Data

	Original Site	Redesigned Site
Average Task Completion Rate	72%	95%
Average Task Time (mins)	2.2	0.84
Average Efficiency	33%	113%

the redesigned site and the task completion rates were significantly higher. Figure 9.7 shows the task times for the original and redesigned sites. Table 9.13 shows a summary of the task completion rates and task times for both versions of the site, as well as an overall measure of efficiency for both (task completion rate per unit time).

So far, everything is very straightforward and simply illustrates some of the usability metrics we've discussed in this book. But here's where it gets interesting. To begin calculating ROI from the changes made to the site, Withrow et al. (2000) made the following assumptions and calculations related to the *time savings*:

- Of the 2.7 million residents of the state, we might "conservatively estimate" a quarter of them use the website at least once per month.

- If each of them save 79 seconds (as was the average task savings in this study), then about 53 million seconds (14,800 hours) are saved per year.
- Converting this to labor costs, we find 370 person-weeks (at 40 hours per week) or 7 person-years are saved per month. 84 person-years are saved each year.
- On average, a citizen in the target state has an annual salary of $14,700.
- This leads to a yearly benefit of *$1.2 million* based only on the time savings.

Notice that this chain of reasoning had to start with a pretty big assumption: that a quarter of the residents of the state use the site at least once per month. So that assumption, which all the rest of the calculations hinge on, is certainly up for debate. A better way of generating an appropriate value with which to start these calculations would have been from actual usage data for the current site.

Then they went on to calculate an increase in revenue due to the increased task completion rate for the new site:

1. The task failure rate of the old portal was found to be 28 percent, whereas the new site was 5 percent.
2. We might assume that 100,000 users would pay a service fee on the order of $2 per transaction at least once a month.
3. Then the 23 percent of them who are succeeding on the new site, whereas formerly they were failing, are generating an additional $552,000 in revenue per year.

Again, a critical assumption had to be made early in the chain of reasoning: that 100,000 users would pay a service fee to the state on the order of $2 per transaction at least once a month.

A better way of doing this calculation would have been to use data from the live site specifically about the frequency of fee-generating transactions (and the amounts of the fees). These could then have been adjusted to reflect the higher task completion rate for the redesigned site. If you agree with their assumptions, these two sets of calculations yield a total of about $1.75 million annually, either in time savings to the residents or in increased fees to the state.

This example points out some of the challenges associated with calculating usability ROI. In general, there are two major classes of situations where you might try to calculate a usability ROI: when the users of the product are employees of your company and when the users of the product are your customers. It tends to be much more straightforward to calculate ROI when the users are employees of your company. You generally know how much the employees are paid, so time savings in completing certain tasks (especially highly repetitive ones) can be directly translated to dollar savings. In addition, you may know the costs involved in correcting certain types of errors, so reductions in the rates of those errors could also be translated to dollar savings.

Calculating usability ROI tends to be much more challenging when the users are your customers (or really anyone not an employee of your company). Your benefits are much more indirect. For example, it might not make any real difference to your bottom line that your customers can complete a key income-generating transaction in 30 percent less time than before. It probably does *not* mean that they will then be performing significantly more of those transactions. But what it *might* mean is that over time those customers will remain your customers and others will become your customers who might not have otherwise (assuming the transaction times are significantly shorter than they are for your competitors), thus increasing revenue. A similar argument can be made for increased task completion rates.

9.5 SIX SIGMA

Six Sigma is a business methodology focused on measuring quality improvement. *Sigma* refers to the standard deviation, so *Six Sigma* refers to six standard deviations. In a manufacturing process, Six Sigma is equivalent to 3.4 defects per 1 million parts manufactured. Six Sigma, as a process improvement methodology, was originated by Bill Smith at Motorola (Motorola, 2007). Since then, the methodology has been adopted by a number of companies, and many books have been written on the topic (e.g., Gygi et al., 2005; Pande & Holpp, 2001; Pyzdek, 2003).

The basic concept underlying Six Sigma is illustrated in Figure 9.8. By going 3 standard deviations (σ, or sigma) above and below the mean, you account for about 99.7 percent of the cases. That leaves only 0.3 percent of the cases outside of that range. The basic goal of Six Sigma is to achieve that level of quality—that the frequency of defects in whatever you're dealing with is at the level of only 0.3 percent or lower. But in many cases, it's not really possible to exactly measure the

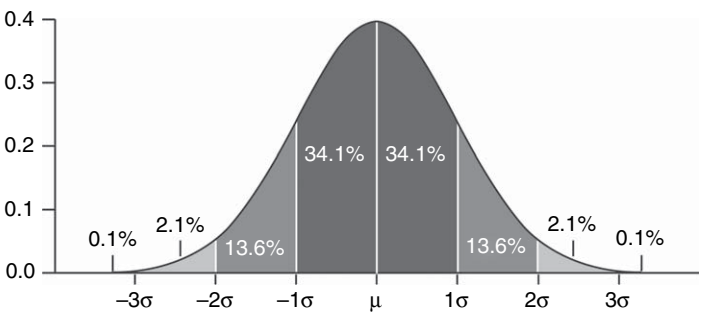


FIGURE 9.8

The normal distribution, illustrating the fact that ±3 standard deviations (i.e., Six Sigma) account for about 99.7 percent of the cases. *Source:* From Wikimedia Commons, based on an original graph by Jeremy Kemp.

frequency of defects. Instead, the goal is often to achieve a Six-Sigma improvement in whatever you are measuring.

So how does any of this apply to usability data? Sigma can certainly be calculated for most of the metrics we've discussed (e.g., task completion, time, errors, satisfaction). At least one interpretation of how Six Sigma could be applied to usability metrics is that you want to try to achieve improvement in a usability metric, across iterations, that's at least three standard deviations higher (better) than where you started. But is that possible with usability data?

Let's look at the data from LeDoux, Mangan, and Tullis (2005), who reported the results of two usability tests of an intranet site: one before a redesign of the site and the other after. Figure 9.9 shows the distribution of SUS scores for the two versions of the site. For the original site, the mean SUS score was 49.1, with a standard deviation of 6.25. Three times the standard deviation would be 18.8, meaning that the "critical mean SUS score" to beat would be 49.1 + 18.8, or 67.9. For the redesigned site, the mean SUS score was 81.8, with a standard deviation of 13.54. The standard error of the mean was 3.50, which means that the lower limit of the 99.7 percent confidence interval for the mean would be 71.4. Since that

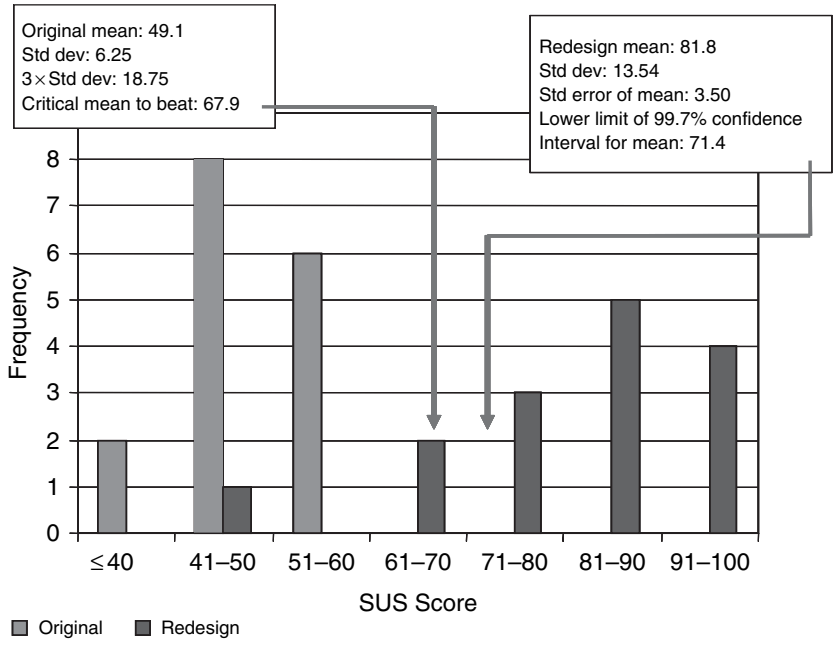


FIGURE 9.9

Data with annotations to illustrate the Six-Sigma improvement in the SUS score from the original version to the redesigned version. *Source:* Adapted from LeDoux, Connor, and Tullis (2005); used with permission.

lower limit is higher than the critical mean to beat from the original version (67.9), you can say that the redesign achieved a Six-Sigma improvement in the SUS rating.

So this is at least one illustration that a Six-Sigma improvement in a usability metric is possible. Is it common to achieve this much improvement in a usability metric in just one design iteration? Probably not. But nobody ever said Six Sigma was easy. Achieving it over multiple iterations may be much more realistic.

9.6 SUMMARY

Here are some of the key takeaways from this chapter:

1. If you're dealing with a live website, you should be studying what your users are doing on the site as much as you can. Don't just look at page hit counts. Look at click-through rates and drop-off rates. Whenever possible, conduct live A/B tests to compare alternative designs (typically with small differences). Use appropriate statistics (e.g., chi-square) to make sure any differences you're seeing are statistically significant.
2. Card-sorting can be immensely helpful in learning how to organize some information or an entire website. Consider starting with an open sort and then following up with one or more closed sorts. Hierarchical cluster analysis and multidimensional scaling (MDS) are useful techniques for summarizing and presenting the results. Closed card-sorts can be used to compare how well different information architectures work for the users.
3. Accessibility is just usability for a particular group of users. Whenever possible, try to include older users and users with various kinds of disabilities in your usability tests. In addition, you should evaluate your product against published accessibility guidelines or standards, such as WCAG or Section 508.
4. Calculating ROI data for usability work is sometimes challenging, but it usually can be done. If the users are employees of your company, it's generally easy to convert usability metrics like reductions in task times into dollar savings. If the users are external customers, you generally have to extrapolate usability metrics, like improved task completion rates or improved overall satisfaction, to decreases in support calls, increases in sales, or increases in customer loyalty.
5. One way of applying Six Sigma concepts to usability data is to see if you can achieve a Six-Sigma improvement in a given usability metric (e.g., task completion rates, task times, satisfaction ratings) from one design iteration or release of a product to another. If so, you can claim that you're moving toward the goals of Six Sigma.