MEASURING THE USER EXPERIENCE

Collecting, Analyzing, and Presenting Usability Metrics

TOM TULLIS
BILL ALBERT

MK
MORGAN KAUFMANN

CHAPTER

Performance Metrics

4

Anyone who uses technology has to interact with some type of interface to accomplish his or her goals. For example, a user of a website clicks on different links, a user of a word-processing application enters information via a keyboard, and a user of a DVD player pushes buttons on a remote control. No matter the technology, users are behaving or interacting with a product in some way. These behaviors form the cornerstone of performance metrics.

Every type of user behavior is measurable in some way. For example, you can measure whether users clicking through a website found what they were looking for. You can measure how long it took users to enter and properly format a page of text in a word-processing application or how many incorrect buttons users pressed in trying to play a DVD. All performance metrics are calculated based on specific user behaviors.

Performance metrics rely not only on user behaviors but also on the use of scenarios or tasks. For example, if you want to measure success, the user needs to have specific tasks or goals in mind. The task may be to find the price of a CD or submit an expense report. Without tasks, performance metrics aren't possible. You can't measure success if the user is only aimlessly browsing a website or playing with a piece of software. How do you know if he or she was successful?

Performance metrics are among the most valuable tools for any usability professional. They're the best way to evaluate the effectiveness and efficiency of many different products. If users are making many errors, you know there are opportunities for improvement. If users are taking four times longer to complete a task than what was expected, efficiency can be greatly improved. Performance metrics are the best way of knowing how well users are actually using a product.

Performance metrics are also useful to estimate the *magnitude* of a specific usability issue. Many times it is not enough to know that a particular issue exists. You probably want to know *how many* people are likely to encounter the same issue after the product is released. For example, by calculating a success rate that includes a confidence interval, you can derive a reasonable estimate of how big a usability issue really is. By measuring task completion times, you can determine what percentage of your target audience will be able to complete a task within a specified amount of time. If only 20 percent of the target users are successful at a particular task, it should be fairly obvious that the task has a usability problem.

Senior managers and other key stakeholders on a project usually sit up and pay attention to performance metrics, especially when they are presented effectively. Managers will want to know how many users are able to successfully complete a core set of tasks using a product. They see these performance metrics as a strong indicator of overall usability and a potential predictor of cost savings or increases in revenue.

Performance metrics are not the magical elixir for every situation. Similar to other metrics, an adequate sample size is required. Although the statistics will work whether you have 2 or 100 participants, your confidence level will change dramatically depending on the sample size. If you're only concerned about identifying the lowest of the low-hanging fruit, performance metrics are probably not a good use of time or money. But if you have the time to collect data from at least eight participants, and ideally more, you should be able to derive meaningful performance metrics with reasonable confidence levels.

Overrelying on performance metrics may be a danger for some. When reporting task success or completion time, it may be easy to lose sight of the underlying issues behind the data. Performance metrics tell the *what* very effectively but not the *why*. Performance data can point to tasks or parts of an interface that were particularly problematic for participants, but you will usually want to supplement with other data, such as observational or self-reported data, to better understand why they were problems and how they might be fixed.

Five basic types of performance metrics are covered in this chapter.

1. *Task success* is perhaps the most widely used performance metric. It measures how effectively users are able to complete a given set of tasks. Two different types of task success will be reviewed: binary success and levels of success.
2. *Time-on-task* is a common performance metric that measures how much time is required to complete a task.
3. *Errors* reflect the mistakes made during a task. Errors can be useful in pointing out particularly confusing or misleading parts of an interface.
4. *Efficiency* can be assessed by examining the amount of effort a user expends to complete a task, such as the number of clicks in a website or the number of button presses on a cell phone.
5. *Learnability* is a way to measure how performance changes over time.

4.1 TASK SUCCESS

The most common usability metric is task success, which can be calculated for practically any usability study that includes tasks. It's almost a universal metric because it can be calculated for such a wide variety of *things* being tested—from websites to kitchen appliances. As long as the user has a well-defined task, you can measure success.

Task success is something that almost anyone can relate to. It doesn't require elaborate explanations of measurement techniques or statistics to get the point across. If your participants can't complete their tasks, then you know something is wrong. Seeing participants fail to complete a simple task can be pretty compelling evidence that something needs to be fixed.

4.1.1 Collecting Any Type of Success Metric

To measure task success, each task that participants are asked to perform must have a clear end-state, such as purchasing a product, finding the answer to a specific question, or completing an online application form. To measure success, you need to know what constitutes success, so you should define the success criteria for each task prior to the data collection. If you don't predefine the criteria, you run the risk of constructing a poorly worded task and not collecting clean success data. Here are examples of two tasks with clear and not-so-clear end-states:

- Find the current price for a share of Google stock (clear end-state)
- Research ways to save for your retirement (not a clear end-state)

Although the second task may be perfectly appropriate in certain types of usability studies, it's not appropriate for measuring task success.

The most common way of measuring success in a lab-based usability test is to have the participant verbally articulate the answer after completing the task. This is natural for the participant, but sometimes it results in answers that are difficult to interpret. Participants might give extra or arbitrary information that makes it difficult to interpret the answer. In these situations, you may need to probe the participants to make sure they actually completed the task successfully.

Another way to collect success data is by having participants provide their answers in a more structured way, such as using an online tool or paper form. Each task might have a set of multiple-choice responses. Participants might choose the correct answer from a list of four to five distracters. It's important to make the distracters as realistic as possible. Try to avoid write-in answers if possible. It's much more time consuming to analyze each write-in answer, and it may involve judgment calls, thereby adding more noise to the data.

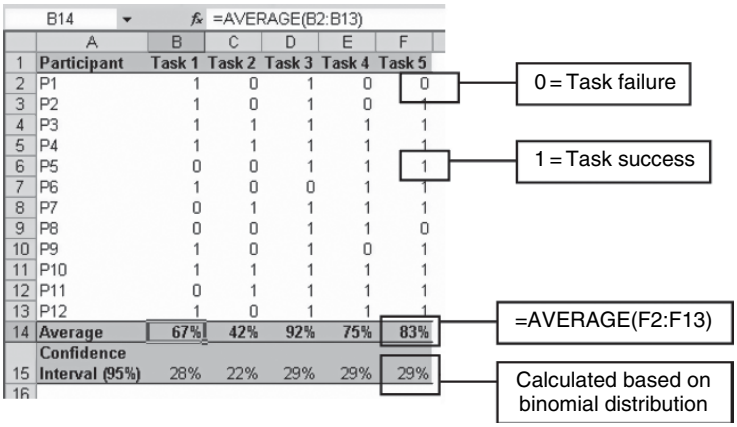
In some cases the correct solution to a task may not be verifiable because it depends on the user's specific situation, and testing is not being performed in person. For example, if you ask participants to find the balance in their savings account, there's no way to know what that amount really is unless you're sitting next to them while they do it. So in this case, you might use a proxy measure of success. For example, you could ask the participant to identify the title of the page that shows the balance. This works well as long as the title of the page is unique and obvious and you're confident that they are able to actually see the balance if they reach this page.

4.1.2 Binary Success

Binary success is the simplest and most common way of measuring task success. Either participants complete a task successfully or they don't. It's kind of like a "pass/fail" course in college. Binary success is appropriate to use when the success of the product depends on users completing a task or a set of tasks. Getting close doesn't count. The only thing that matters is that they accomplish their tasks. For example, when evaluating the usability of a defibrillator device (to resuscitate people during a heart attack), the only thing that matters is being able to use it correctly without making any mistakes. Anything less would be a major problem, especially for the recipient! A less dramatic example might be a task that involves purchasing a book on a website. Although it may be helpful to know where in the process someone failed, if your company's revenue depends on selling those books, that's what really matters.

How to Collect and Measure Binary Success

Each time users perform a task, they should be given a "success" or "failure" score. Typically, these scores are in the form of 1's (for success) and 0's (for failure). (The analysis is easier if you assign a numeric score rather than a text value of "success" or "failure.") By having a numeric score, you can easily calculate the average as well as perform any other statistics you might need. Simply calculate the average of the 1's and 0's to determine the binary success rate. In addition to showing the average, it's best to also include the confidence interval as part of the binary success data. Figure 4.1 shows how to organize binary success data. Assuming the answers are predefined, it should be very easy to assign a success or failure



	A	B	C	D	E	F
1	Participant	Task 1	Task 2	Task 3	Task 4	Task 5
2	P1	1	0	1	0	0
3	P2	1	0	1	0	1
4	P3	1	1	1	1	1
5	P4	1	1	1	1	1
6	P5	0	0	1	1	1
7	P6	1	0	0	1	1
8	P7	0	1	1	1	1
9	P8	0	0	1	1	0
10	P9	1	0	1	0	1
11	P10	1	1	1	1	1
12	P11	0	1	1	1	1
13	P12	1	0	1	1	1
14	Average	67%	42%	92%	75%	83%
15	Confidence Interval (95%)	28%	22%	29%	29%	29%
16						

FIGURE 4.1

An example of how binary success data should be organized. Confidence intervals were calculated based on a binomial distribution.

score for each user on each task. If you are unsure whether to count an answer as a task success or failure, it's critical to look at the degree to which the task was accomplished by obtaining that answer.

How to Analyze and Present Binary Success

The most common way to analyze and present binary success rates is by individual task. This involves simply presenting the percentage of participants who successfully completed each task (Figure 4.2). This approach is most useful when you want to compare success rates for tasks. You can then do a more detailed analysis of each task by looking at specific problems to determine what changes may be needed to address them. If you're interested in knowing whether there is a statistically significant difference between the various tasks, you will need to perform a *t*-test or ANOVA (see Chapter 2). In the figure, the average binary success rate for task 1 is 67 percent, but there is a 95 percent chance that the true mean is between 39 and 95 percent (as shown by the confidence interval).

Another common way of looking at binary success is by user or type of user. As always in reporting usability data, you should be careful to maintain the anonymity of the participants in the study by using numbers or other nonidentifiable descriptors. The main value of looking at binary success data from a user perspective is that you can identify different groups of participants who perform differently or encounter different sets of problems. Here are some of the common ways to segment different participants:

- Frequency of use (infrequent users versus frequent users)
- Previous experience using the product
- Domain expertise (low-domain knowledge versus high-domain knowledge)
- Age group

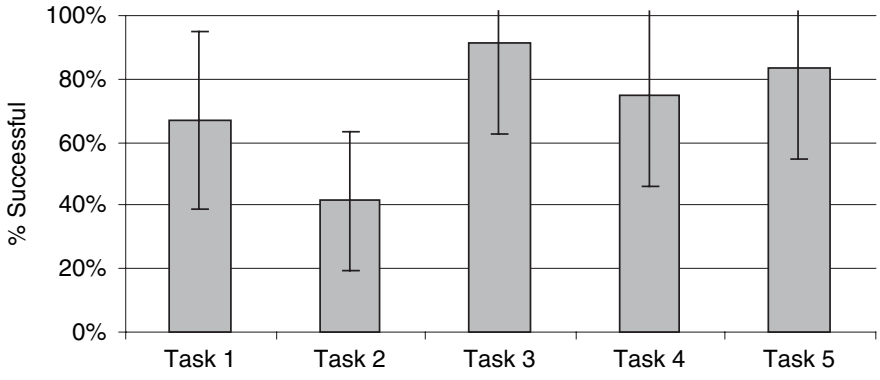


FIGURE 4.2

An example of how to present binary success data for individual tasks. The error bars represent the 95 percent confidence interval based on a binomial distribution.

One of the advantages of looking at success data by participant is that you can then calculate a percentage of tasks that each successfully completed, assuming everyone was given more than one task. In this way, the data is no longer binary, and it becomes continuous. For example, if you are testing an application for use by both managers and non-managers, and you find that six out of six managers failed to complete a given task, but none of the six non-managers failed the task, you should investigate why that happened. There may be something in the product for that particular task that works well for the non-managers but is confusing for the managers. Of course, you must be careful drawing conclusions about different groups of participants based on small sample sizes.

If you have a fairly large number of participants in a usability study (at least 12, ideally more than 20), it may be helpful to present binary success data as a frequency distribution (Figure 4.3). This is a convenient way to visually represent the variability in binary task success data. For example, in Figure 4.3, six participants in the evaluation of the original website completed 61 to 70 percent of the tasks successfully, one completed fewer than 50 percent, and two completed as many as 81 to 90 percent. In a revised design, six participants had a success rate of 91 percent or greater, and no participant had a success rate below 61 percent. Illustrating that the two distributions of task success barely overlap is a much more dramatic way of showing the improvement across the iterations than simply reporting the two means.

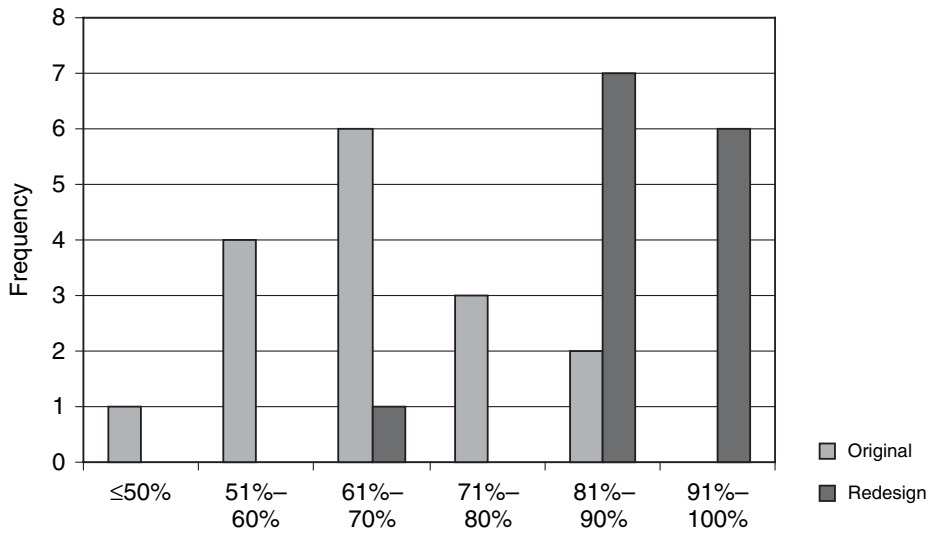


FIGURE 4.3

Frequency distributions of binary success rates from usability tests of the original version of a website and the redesigned version. *Source:* Adapted from LeDoux, Connor, and Tullis (2005); used with permission.

Calculating Confidence Intervals for Binary Success

One of the most important aspects of analyzing and presenting binary success is including confidence intervals. Confidence intervals are essential because they reflect your trust or confidence in the data. In most usability studies, binary success data are based on relatively small samples (e.g., 5–20 participants). Consequently, the binary success metric may not be as reliable as we would like it to be. For example, if 4 out of 5 participants successfully completed a task, how confident can we be that 80 percent of the larger population of participants will be able to successfully complete that task? Obviously, we would be more confident if 16 out of 20 participants successfully completed the task and even more confident if 80 out of 100 did.

Fortunately, there is a way to take this into account. Binary success rates are essentially proportions: the proportion of the participants who successfully completed a given task. The appropriate way to calculate a confidence interval for a proportion like this is to use is a binomial confidence interval. Several methods are available for calculating binomial confidence intervals, such as the Wald Method and the Exact Method. But as Sauro and Lewis (2005) have shown, many of those methods are too conservative or too liberal in their calculation of the confidence interval when you're dealing with the small sample sizes we commonly have in usability tests. They found that a modified version of the Wald Method, called the Adjusted Wald, yielded the best results when calculating a confidence interval for task success data.

Jeff Sauro has provided a very useful calculator for determining confidence intervals for binary success on his website: *http://www.measuringusability.com/wald.htm*. By entering the total number of participants who attempted a given task and how many of them successfully completed it, this tool will automatically perform the Wald, Adjusted Wald, Exact, and Score calculations of the confidence interval for the mean task completion rate. You can choose to calculate a 99, 95, or 90 percent confidence interval. If you really want to calculate confidence intervals for binary success data yourself, the details are included on our website: *www.MeasuringUserExperience.com*.

If 4 out of 5 participants successfully completed a given task, the Adjusted Wald Method yields a 95 percent confidence interval for that task completion rate ranging from 36 to 98 percent—a rather large range! On the other hand, if 16 out of 20 participants completed the task successfully (the same proportion), the Adjusted Wald Method yields a 95 percent confidence interval of 58 to 93 percent. If you got *really* carried away and ran a usability test with 100 participants, of whom 80 completed the task successfully, the 95 percent confidence interval would be 71 to 87 percent. As is always the case with confidence intervals, larger sample sizes yield smaller (or more accurate) intervals.

4.1.3 Levels of Success

Identifying levels of success is useful when there are reasonable shades of gray associated with task success. The participant receives some value from partially

completing a task. Think of it as partial credit on a homework assignment if you showed your work, even though you got the wrong answer. For example, assume that a participant's task is to find the cheapest digital camera with at least 5 megapixel resolution, at least 6X optical zoom, and weighing no more than 3 pounds. What if the participant found a camera that met most of those criteria but had a 5X optical zoom instead of a 6X?

According to a strict binary success approach, that would be a failure. But you're losing some important information by doing that. The participant actually came very close to successfully completing the task. In some cases this might be acceptable. For some types of products, coming close to fully completing a task may provide value to the participant. Also, it may be helpful for you to know why some participants failed to complete a task or with which particular tasks they needed help.

How to Collect and Measure Levels of Success

Collecting and measuring levels of success data are very similar to collecting and measuring binary success data, except that you must define the various levels. There are three perspectives on levels of success:

- Levels of success might be based on the extent or degree to which a participant completed the task. This could be defined by whether he received any assistance or got only part of an answer.
- Levels of success might be based on the experience in completing a task. Some participants might struggle, whereas others can complete their tasks without any difficulty.
- Levels of success might be based on the participant accomplishing the task in different ways. Some participants might accomplish the task in an optimal way, whereas others might accomplish it in ways that are less than optimal.

Between three and six levels of success based on the degree to which participants complete a task are typical. A common approach is to use three levels: complete success, partial success, and complete failure.

Levels of success data are almost as easy to collect and measure as binary success data. It just means defining what you mean by ''complete success'' and by ''complete failure.'' Anything in between is considered a partial success. A more granular approach is to break out each level according to whether assistance was given or not. The following are examples of six different levels of completion:

- Complete success
  - With assistance
  - Without assistance
- Partial success
  - With assistance
  - Without assistance

- Failure
  - Participant thought it was complete, but it wasn't
  - Participant gave up

If you do decide to use levels of success, it's important to clearly define the levels beforehand. Also, consider having multiple observers independently assess the levels for each task and then reach a consensus.

A common issue when measuring levels of success is deciding what constitutes "giving assistance" to the participant. Here are some examples of situations we define as giving assistance:

- Moderator takes the participant back to a homepage or resets to an initial (pretask) state. This form of assistance may reorient the participant and help avoid certain behaviors that initially resulted in confusion.
- Moderator asks probing questions or restates the task. This may cause the participant to think about her behavior or choices in a different way.
- Moderator answers a question or provides information that helps the participant complete the task.
- Participant seeks help from an outside source. For example, the participant calls a phone representative, uses some other website, consults a user manual, or accesses an online help system.

Levels of success based on completion can be organized in many different ways. One of the most straightforward methods is to assign a numeric value for each level. Here is one way of assigning weights for various levels of success:

- Complete success (without assistance) = 1.0
- Partial success = 0.5
- Gives up or wrong answer = 0.0

This method allows you to derive a usability "success score," and you can easily calculate an average. This is not a traditional success rate but a success score. One obvious limitation is that it does not differentiate between different types of failure (giving up versus wrong answers). So if it's important to know the difference between participants who gave the wrong answer and those who just gave up, numeric scoring may not be the best solution.

Level of success can also be examined in terms of the user experience. We commonly find that some tasks are completed without any difficulty and others are completed with minor or major problems along the way. It's important to distinguish between these different experiences. A 4-point scoring method can be used for each task:

1 = *No problem.* The participant successfully completed the task without any difficulty or inefficiency.

2 = *Minor problem*. The participant successfully completed the task but took a slight detour. He made one or two small mistakes but quickly recovered and was successful.

3 = *Major problem.* The participant successfully completed the task but had major problems. She struggled and took a major detour in her eventual successful completion of the task.

4 = *Failure/gave up.* The participant provided the wrong answer or gave up before completing the task, or the moderator moved on to the next task before successful completion.

When using this scoring system, it's important to remember that these data are ordinal (section 2.2.2). Therefore, you should not report an average score. Rather, present the data as frequencies for each level of completion. This scoring system is relatively easy to use, and we usually see agreement on the various levels by different usability specialists observing the same interactions. Also, you can aggregate the data into a binary success rate if you need to. Finally, this scoring system is usually easy to explain to your audience.

Another way of organizing levels of success data is according to different answers given by participants. For example, you could assign a score of 1.0 for an optimal answer(s) and a score or 0.75 or 0.5 for acceptable (but suboptimal) answers, depending on the quality of the answer. You don't have to assign a numeric score, but it's convenient if you want to do any in-depth analysis. The same can also be done for different navigation strategies used to reach a particular answer. If you are evaluating a website, sometimes the participant clicks on the best link and completes the task without any issues. Other times a participant might click on a suboptimal link but eventually be successful.

How to Analyze and Present Levels of Success

In analyzing levels of success, the first thing you should do is create a stacked bar chart. This will show the percentage of participants who fall into each category or level, including failures. Make sure that the bars add up to 100 percent. Figure 4.4 is an example of a common way to present levels of success.

Another approach to analyzing and presenting levels of success based on task completion is to report a "usability score." As mentioned in the previous section, you can assign a numeric score to each level of success ranging from 0 (failure) to 1.0 (full success without assistance). Obviously there are many different ways to do this. These data would look very much like binary success (Figure 4.2), except instead of the *y*-axis showing "% successful," it would be "average success score."

Keep in mind that it's important to communicate the scoring system used to the audience, because it's easy to misinterpret these charts as common success rates.

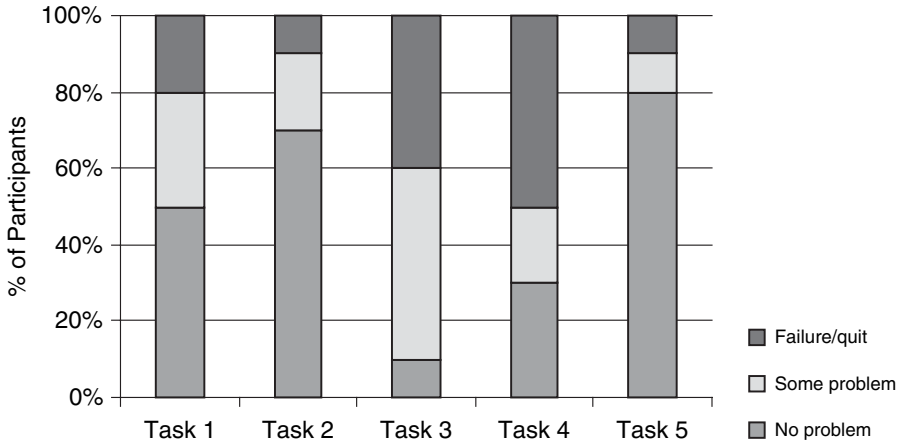


FIGURE 4.4

Stacked bar chart showing different levels of success based on task completion.

4.1.4 Issues in Measuring Success

Obviously, an important issue in measuring task success is simply how you define whether a task was successful. The key is to clearly define beforehand what the criteria are for *successfully* completing each task. Try to think through the various situations that might arise for each task and decide whether they constitute success. For example, is a task successful if the participant finds the right answer but reports it in the wrong format? Also, what happens if he reports the right answer but then restates his answer incorrectly? When unexpected situations arise during the test, make note of them and afterward try to reach a consensus among the observers about those cases.

One issue that commonly arises during a usability evaluation is how or when to end a task if the participant is not successful. In essence, this is the "stopping rule" for unsuccessful tasks. Here are some of the common approaches to ending an unsuccessful task:

1. Tell the participants at the beginning of the session that they should continue to work on each task until they either complete it or reach the point at which, in the real world, they would give up or seek assistance (from technical support, a colleague, etc.).

2. Apply a "three strikes and you're out" rule. This means that the participants make three attempts to complete a task before you stop them. The main difficulty with this approach is defining what is meant by an "attempt." It could be three different strategies, three wrong answers, or three different "detours" in finding specific information. However you define it, there will be a considerable amount of discretion on behalf of the moderator or scorer.

3. "Call" the task after a prespecified amount of time has passed. Set a time limit, such as five minutes. After the time has expired, move on to the next task. In most cases, it is better not to tell the participant that you are timing him. By doing so, you create a more stressful, "testlike" environment.

Of course, you always have to be sensitive to the participant's state in any usability session and potentially end a task (or even the session) if you see that the participant is becoming particularly frustrated or agitated.

4.2 TIME-ON-TASK

Time-on-task (sometimes referred to as task completion time or simply task time) is an excellent way to measure the efficiency of any product. The time it takes a participant to perform a task says a lot about the usability of the product. In almost every situation, the faster a participant can complete a task, the better the experience. In fact, it would be pretty unusual for a user to complain that a task took less time than expected.

There are a couple of exceptions to the assumption that faster is better. One is a game where you may not want the participant to finish it too quickly. The main purpose of most games is the experience itself rather than the quick completion of a task. Another exception may be learning. For example, if you're putting together an online training course, slower may be better. It may be better that participants not rush through the course but spend more time completing their tasks.

4.2.1 Importance of Measuring Time-on-Task

Time-on-task is particularly important for products where tasks are performed repeatedly by the user. For example, if you're designing an application for use by customer service representatives of an airline, the time it takes to complete a phone reservation would be an important measure of efficiency. The faster the airline agent can complete a reservation, presumably the more calls can be handled and, ultimately, the more money can be saved. The more often a task is performed by the same participant, the more important efficiency becomes. One of the side benefits of measuring time-on-task is that it can be relatively straightforward to calculate cost savings due to an increase in efficiency and then derive an actual ROI (return on investment). Calculating ROI is discussed in more detail in section 9.4.

4.2.2 How to Collect and Measure Time-on-Task

Time-on-task is simply the time elapsed between the start of a task and the end of a task, usually expressed in minutes and seconds. Logistically, time-on-task can be measured in many different ways. The moderator or note taker can use a stopwatch or any other time-keeping device that can measure at the minute and second levels. Using a digital watch, you could simply record the start and end times. When

videotaping a usability session, we find it's helpful to use the time-stamp feature of most recorders to display the time and then to mark those times as the task start and stop times. If you choose to record time-on-task manually, it's important to be very diligent about when to start and stop the clock and/or record the start and stop times. It may also be helpful to have two people record the times.

Automated Tools for Measuring Time-on-Task

A much easier and less error-prone way of recording task times is using an automated tool. Usability testing products such as Ergo Browser, Data Logger, or Bailey's Usability Testing Environment (UTE) all capture task time automatically. In fact, tools such as UTE will calculate average task completion times for you. The automated method has several advantages. Not only is it less error-prone but it's also much less obtrusive. The last thing you want is a participant in a usability session to feel nervous from watching you press the start and stop button on your stopwatch.

WORKING WITH TIME DATA IN EXCEL

If you use Excel to log data during a usability test, it's often convenient to use times that are formatted as hours, minutes, and (sometimes) seconds (hh:mm:ss). Excel provides a variety of formats for time data. This makes it easy to enter the times, but it slightly complicates matters when you need to calculate an elapsed time. For example, assume that a task started at 12:46 PM and ended at 1:04 PM. Although you can look at those times and determine that the elapsed time was 18 minutes, how to get Excel to calculate that isn't so obvious. Internally, Excel stores all times as a number reflecting the number of seconds elapsed since midnight. So to convert an Excel time to minutes, multiply it by 60 (the number of minutes in an hour) and then by 24 (the number of hours in a day). To convert to seconds, multiply by another 60 (the number of seconds in a minute).

Turning on and off the Clock

Not only do you need a way to measure time, but you also need some rules about *how* to measure time. Perhaps the most important rule is when to turn the clock on and off. Turning on the clock is fairly straightforward: If you have the participants read the task aloud, you start the clock as soon as they finish reading the task.

Turning off the clock is a more complicated issue. Automated time-keeping tools typically have an ''answer'' button. Participants are required to hit the ''answer'' button, at which point the timing ends, and they are asked to provide an answer and perhaps answer a few questions. If you are not using an automated method, you can have participants verbally report the answer or perhaps even write it down. However, there are many situations in which you may not be sure if they have found the answer. In these situations, it's important for participants to indicate their answer as quickly as possible. In any case, you want to stop timing when the participant has stopped interacting with the product. Because this is largely a matter of interpretation, the data might contain a fair bit of noise.

Tabulating Time Data

The first thing you need to do is arrange the data in a table, as shown in Table 4.1. Typically, you will want a list of all the participants in the first column, followed by the time data for each task in the remaining columns (expressed in seconds, or in minutes if the tasks are long). Table 4.1 also shows summary data, including the average, median, geometric mean, and confidence intervals for each task.

Table 4.1 Time-on-Task Data for 20 Participants and 5 Tasks

Participant	Task 1	Task 2	Task 3	Task 4	Task 5
P1	259	112	135	58	8
P2	253	64	278	160	22
P3	42	51	60	57	26
P4	38	108	115	146	26
P5	33	142	66	47	38
P6	33	54	261	26	42
P7	36	152	53	22	44
P8	112	65	171	133	46
P9	29	92	147	56	56
P10	158	113	136	83	64
P11	24	69	119	25	68
P12	108	50	145	15	75
P13	110	128	97	97	78
P14	37	66	105	83	80
P15	116	78	40	163	100
P16	129	152	67	168	109
P17	31	51	51	119	116
P18	33	97	44	81	127
P19	75	124	286	103	236
P20	76	62	108	185	245
Average	**86.6**	**91.5**	**124.2**	**91.35**	**80.3**
Median	**58.5**	**85**	**111.5**	**83**	**66**

Table 4.1 *cont.*

Participant	Task 1	Task 2	Task 3	Task 4	Task 5
Geometric mean	65.216	85.225	104.971	73.196	60.323
Upper bound	119.8	108.0	159.5	116.6	110.2
Lower bound	53.4	75.0	119.9	66.1	50.4
Confidence interval	33.2	16.5	19.8	25.2	29.9

Note: Data are all expressed in seconds.

4.2.3 Analyzing and Presenting Time-on-Task Data

You can analyze and present time-on-task data in many different ways. Perhaps the most common way is to look at the average amount of time spent on any particular task or set of tasks by averaging all the times for each participant by task (Figure 4.5). This is a straightforward and intuitive way to report time-on-task data. One downside is the potential variability across participants. For example, if you have several participants who took an exceedingly long time to complete a task, it may increase the average considerably. Therefore, you should always report a confidence interval to show the variability in the time data. This will not only show the

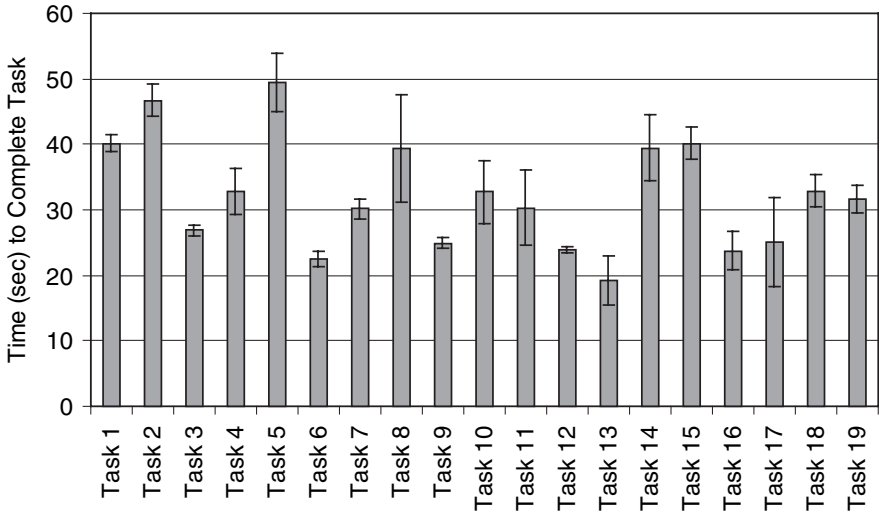


FIGURE 4.5

Mean time-on-task for 19 tasks. Error bars represent a 95 percent confidence interval. These data are from an online study of a prototype website.

variability within the same task but also help visualize the difference across tasks to determine whether there is a statistically significant difference between tasks.

Some usability specialists prefer to summarize time-on-task data using the median rather than the mean. The median is the middle point in an ordered list of all the times: Half of the times are below the median and half are above the median. Similarly, some usability specialists suggest that the geometric mean is potentially less biased. Time data is typically skewed, in which case geometric means may be more appropriate. These alternatives can be calculated in Excel using the `=MEDIAN` or `=GEOMEAN` functions. In practice, we find that using these other methods of summarizing the time data may change the overall level of the times, but the kinds of patterns you're interested in (e.g., comparisons across tasks) usually stay the same; the same tasks still took the longest or shortest times overall.

Ranges

A variation on calculating average completion time by task is to create ranges, or discrete time intervals, and report the frequency of participants who fall into each time interval. This is a useful way to visualize the spread of completion times by all participants. In addition, this might be a helpful approach to look for any patterns in the type of participants who fall within certain segments. For example, you may want to focus on those who had particularly long completion times to see if they share any common characteristics.

Thresholds

Another useful way to analyze task time data is by using a threshold. In many situations, the only thing that matters is whether users can complete certain tasks within an acceptable amount of time. In many ways, the average is unimportant. The main goal is to minimize the number of users who need an excessive amount of time to complete a task. The main issue is determining what the threshold should be for any given task. One way is to perform the task yourself, keep track of the time, and then double that number. Alternatively, you could work with the product team to come up with a threshold for each task based on competitive data or even a best guess. Once you have set your threshold, simply calculate the percentage of users above or below the threshold and plot as illustrated in Figure 4.6.

Distributions and Outliers

When analyzing time data, it's critical to look at the distribution. This is particularly true for time-on-task data collected via automated tools (when the moderator is not present). Participants might take a phone call or even go out to lunch in the middle of a task. The last thing you want is to include a task time of two hours among other times of only 15 to 20 seconds when calculating an average! It's perfectly acceptable to exclude outliers from your analysis, and many statistical techniques for identifying them are available. Sometimes we exclude any times that are more than three standard deviations above the mean. Alternatively, we sometimes set up

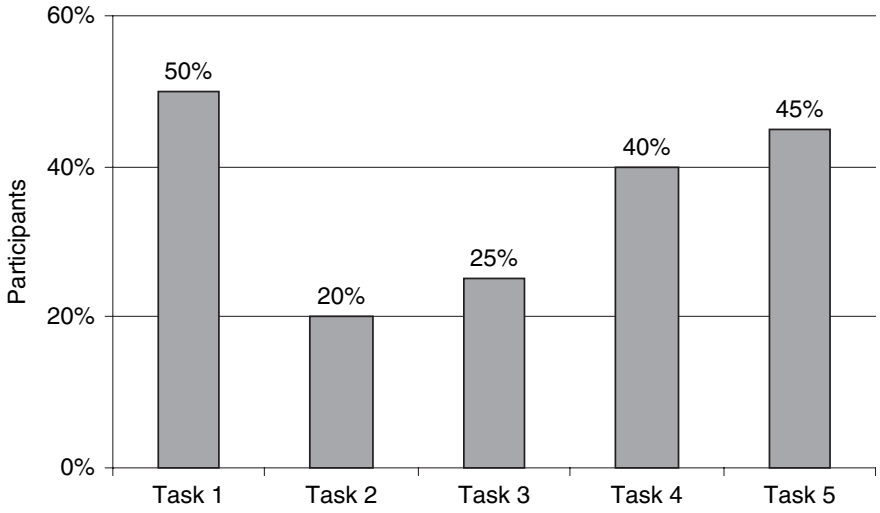


FIGURE 4.6

An example showing the percentage of participants who completed each task in less than one minute.

thresholds, knowing that it should never take a user more than x seconds to complete a task. There is a real art to this last approach. Therefore, you should have some rationale for using an arbitrary threshold for excluding outliers.

The opposite problem—participants apparently completing a task in unusually short amounts of time—is actually more common in online studies. Some participants may be in such a hurry, or only care about the compensation, that they simply fly through the study as fast as they can. In most cases, it's very easy to identify these individuals through their time data. For each task, determine the fastest possible time. This would be the time it would take someone with perfect knowledge and optimal efficiency to complete the task. For example, if there is no way you, as an expert user of the product, can finish the task in less than eight seconds, then it is highly unlikely than a typical participant could complete the task any faster. Once you have established this minimum acceptable time, simply filter out those times that fall below the minimum. You can expect anywhere from 5 to 10 percent of all participants in an online study to be in it only for the compensation!

4.2.4 Issues to Consider When Using Time Data

Some of the issues to think about when analyzing time data is whether to look at all tasks or just the successful tasks, what the impact of using a think-aloud protocol might be, and whether to tell test participants that time is being measured.

Only Successful Tasks or All Tasks?

Perhaps the first issue to consider is whether you should include only successful tasks or all tasks in the analysis. The main advantage of only including successful tasks is that it is a cleaner measure of efficiency. For example, time data for unsuccessful tasks are often very difficult to estimate. Some participants will keep on trying until you practically unplug the computer. Any task that ends with the participant giving up or the moderator "pulling the plug" is going to result in highly variable time data.

The main advantage of analyzing time data for *all* tasks, successful or not, is that it is a more accurate reflection of the overall user experience. For example, if only a small percentage of participants were successful, but that particular group was very efficient, the overall time-on-task is going to be low. Therefore, it is easy to misinterpret time-on-task data when only analyzing successful tasks. Another advantage of analyzing time data for all tasks is that it is an independent measure in relation to the task success data. If you only analyze the time data for successful tasks, you're introducing a dependency between the two sets of data.

A good rule is that if the participant always determined when to give up on unsuccessful tasks, you should include all times in the analyses. If the moderator sometimes decided when to end an unsuccessful task, then use only the times for the successful tasks.

Using a Think-Aloud Protocol

Another important issue to consider is whether to use a think-aloud protocol when collecting time data. Most usability specialists rely heavily on a think-aloud protocol to gain important insight into the user experience. A think-aloud protocol is almost certainly going to have an impact on the time-on-task data. The last thing you want to do is measure time-on-task as a participant is giving a ten-minute diatribe on the importance of fast-loading web pages. A good solution, when you want to capture time-on-task, is to ask participants to "hold" most of their comments for the time between tasks. Then you can have a dialog with the participant about the just-completed task after the "clock is stopped." This is sometimes called a retrospective probing technique (e.g., Birns, Joffre, Leclerc, & Paulsen, 2002).

THINK-ALOUD PROTOCOL AND TASK TIMES

Does using a think-aloud protocol always increase task times? Maybe not. Sometimes it actually might decrease task times. It can be argued that the process of thinking aloud sometimes helps the participant to focus more on the task and the interface, perhaps more readily identifying ways of accomplishing the task.

Should You Tell the Participants about the Time Measurement?

An important question to consider is whether to tell the participants you are recording their time. It's possible that if you don't, participants won't behave in an efficient manner. It's not uncommon for participants to explore different parts

of a website when they are in the middle of a task. On the flip side, if you tell them they are being timed, they may become nervous and feel they are the ones being tested and not the product. A good compromise is to ask participants to perform the tasks as quickly and accurately as possible without volunteering that they are being explicitly timed. If the participants happen to ask (which they rarely do), then simply state that you are noting the start and finish times for each task.

4.3 ERRORS

Some usability professionals believe errors and usability issues are essentially the same thing. Although they are certainly related, they are actually quite different. A usability issue is the underlying *cause* of a problem, whereas one or more errors are a possible *outcome*. For example, if users are experiencing a problem in completing a purchase on an e-commerce website, the issue (or cause) may be confusing labeling of the products. The error, or the result of the issue, may be the act of choosing the wrong options for the product they want to buy. Essentially, errors are incorrect actions that may lead to task failure.

4.3.1 When to Measure Errors

In some situations it's very helpful to identify and classify errors rather than just document usability issues. Measuring errors is useful when you want to understand the specific action or set of actions that may result in task failure. For example, a user may make the wrong selection on a web page and sell a stock instead of buying more. A user may push the wrong button on a medical device and deliver the wrong medication to a patient. In both cases, it's important to know what errors were made and how different design elements may increase or decrease the frequency of errors.

Errors are a useful way of evaluating user performance. While being able to complete a task successfully within a reasonable amount of time is important, the number of errors made during the interaction is also very revealing. Errors can tell you how many mistakes were made, where they were made within the product, how various designs produce different frequencies and types of errors, and generally how usable something really is.

Measuring errors is not right for every situation. We've found that there are three general situations where measuring errors might be useful:

1. When an error will result in a significant loss in efficiency—for example, when an error results in a loss of data, requires the user to reenter information, or significantly slows the user in completing a task.

2. When an error will result in significant costs—for example, if an error will result in increased call volumes to customer support or in increased product returns.
3. When an error will result in task failure—for example, if an error will cause a patient to receive the wrong medication, a voter to accidentally vote for the wrong candidate, or a web user to buy the wrong product.

4.3.2 What Constitutes an Error?

Surprisingly, there is no widely accepted definition of what constitutes an error. Obviously, it's some type of incorrect action on the part of the user. Generally an error is any action that prevents the user from completing a task in the most efficient manner. Errors can be based on many different types of actions by the user, such as the following:

- Entering incorrect data into a form field (such as typing the wrong password during a login attempt)
- Making the wrong choice in a menu or drop-down list (such as selecting "Delete" when they should have selected "Modify")
- Taking an incorrect sequence of actions (such as reformatting their DVD drive when all they were trying to do was play a taped TV show)
- Failing to take a key action (such as clicking on a key link on a web page)

Obviously, the range of possible actions will depend on the product you are studying (website, cell phone, DVD player, etc.). When you're trying to determine what constitutes an error, first make a list of all the possible actions a user can take on your product. Once you have the universe of possible actions, you can then start to define many of the different types of errors that can be made using the product.

A Real-World Example: Errors in Election Ballots

One of the most publicized examples of possible user errors occurred in the 2000 U.S. presidential election in Palm Beach County, Florida. They used the now-infamous "butterfly ballot" shown in Figure 4.7. With this ballot, you record your vote by punching one of the holes in the center strip. The crux of the usability problem is that although Al Gore was the *second* candidate listed on the left, a vote for him was indicated using the *third* hole. The second hole corresponded to Pat Buchanan on the right (no pun intended). How many voters might have accidentally voted for Buchanan when they intended to vote for Gore is not known. Whatever your political leanings, this is a situation where measuring errors in a usability study of the election ballot prior to the actual election would have been very helpful!

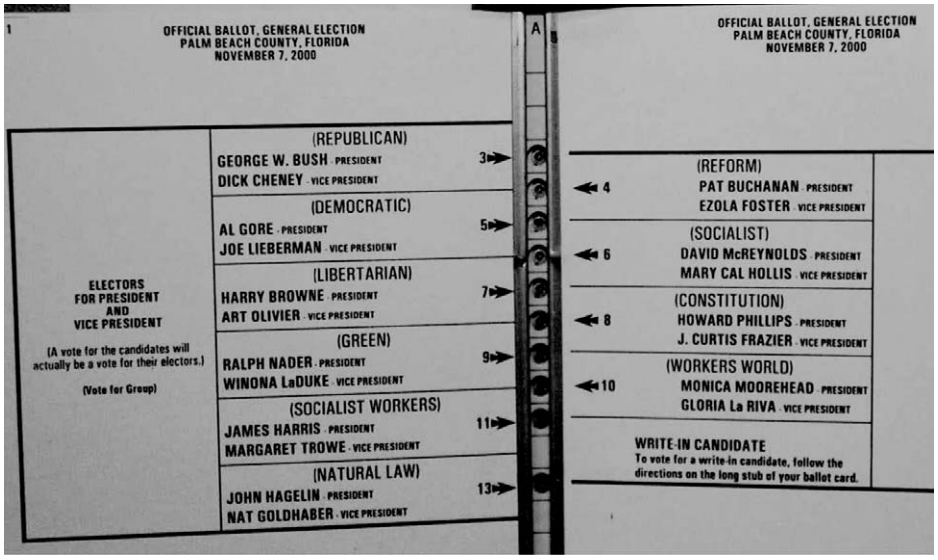


FIGURE 4.7

The ballot used in the 2000 presidential election in Palm Beach County.

4.3.3 Collecting and Measuring Errors

Measuring errors is not always easy. As with other performance metrics, you need to know what the correct action should be or, in some cases, the correct set of actions. For example, if you're studying a password reset form, you need to know what is considered the correct set of actions to successfully reset the password and what is not. The better you can define the universe of correct and incorrect actions, the easier it will be to measure errors.

An important consideration is whether a given task presents only a single error opportunity or multiple error opportunities. An error opportunity is basically a chance to make a mistake. For example, if you're measuring the usability of a login screen, two error opportunities are possible: making an error when entering the username and making an error when entering the password. If you're measuring the usability of an online form, there could be as many error opportunities as there are fields on the form.

In some cases there might be multiple error opportunities for a task but you only care about one of them. For example, you might be interested only in whether users click on a specific link that you know will be critical to completing their task. Even though errors could be made in other places on the page, you're narrowing your scope of interest to that single link. If users don't click on the link, it is considered an error.

The most common way of organizing error data is by task. Simply record the number of errors for each task and each user. If there is only a single opportunity for error, the numbers will be 1's and 0's:

0 = No error 1 = One error

If multiple error opportunities are possible, the numbers will vary between 0 and the maximum number of error opportunities. The more error opportunities, the harder and more time consuming it will be to tabulate the data. You can count errors while observing participants during a lab study, by reviewing videos after the sessions are over, or by collecting the data using an automated or online tool.

4.3.4 Analyzing and Presenting Errors

The analysis and presentation of error data differ slightly depending on whether a task has only one error opportunity or multiple error opportunities.

Tasks with a Single Error Opportunity

The most common way to analyze errors for tasks with single error opportunities is to look at the frequency of the error for each task. This will indicate which tasks are associated with the most errors and thus have the most significant usability issues. This can be done in either of two ways, with slightly different forms of interpretation:

- Run a frequency of errors by task and plot out the number of errors. This would show the number of errors made on each task. Note that you do not need to use confidence intervals in this type of analysis because you are not trying to extrapolate to a more general population; you are only interested in seeing which tasks have the most errors.
- Divide the number of errors by the total number of participants for each task. This will tell you the percentage of participants who made an error for each task. This is especially useful if different numbers of participants performed each task. Figure 4.8 is an example of presenting errors based on a single opportunity. In this example, they were interested in the percentage of participants who experienced an error when using different types of on-screen keyboards (Tullis, Mangan, & Rosenbaum, 2007). The control condition is the current QWERTY keyboard layout.

Another way to analyze and present error metrics for tasks with single error opportunities is from an aggregate perspective. You may not always be concerned about a specific task but about how participants performed overall. Here are some options:

- You could average the error rates for each task into a single error rate. This would tell you the overall error rate for the study. For example, you might be able to say that the tasks had an average error rate of 25 percent. This is a useful bottom-line metric for reporting errors.

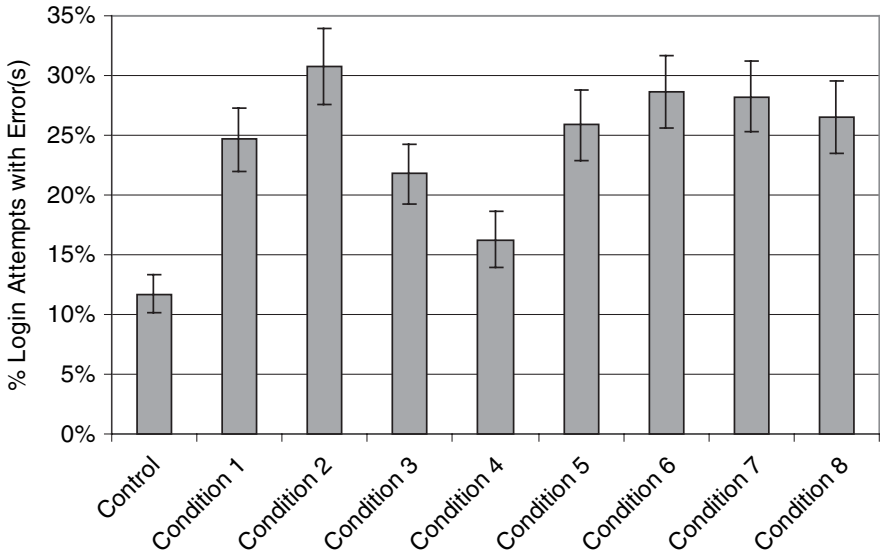


FIGURE 4.8

An example showing how to present data for single error opportunities. In this study, only one error opportunity per task (entering a password incorrectly) was possible, and the graph shows the percentage of participants who made an error for each condition.

- You could take an average of all the tasks that had a certain number of errors. For example, if you are looking at a large number of tasks, you could report that 50 percent of all the tasks had an error rate of 10 percent or greater. Or you might state that at least one participant made an error on 80 percent of the tasks.
- You could establish maximum acceptable error rates for each task. For example, you might only be interested in identifying the tasks that have an error rate above a particular threshold, such as 10 percent. You could then calculate the percentage of tasks above and below this threshold. For example, you might simply state that 25 percent of the tasks exceeded an acceptable error rate.

Tasks with Multiple Error Opportunities

Here are some of the more common ways to analyze the data from tasks that provide multiple error opportunities:

- A good place to start is to look at the frequency of errors for each task. You will be able to see which tasks are resulting in the most errors. But this may be misleading if each task has a different number of error opportunities. In that case, it might be better to divide the total number of errors for the task by the

total number of error opportunities. This creates an error rate that takes into account the number of opportunities.

- You could calculate the average number of errors made by each participant for each task. This will also tell you which tasks are producing the most errors. However, it may be more meaningful because it suggests that a typical user might experience *x* number of errors on a particular task when using the product. Another advantage is that it takes into account the extremes. If you are simply looking at the frequency of errors for each task, some participants may be the source of most of the errors, whereas many others are performing the task error-free. By taking an average number of errors by each participant, this bias is reduced.
- In some situations it might be interesting to know which tasks fall above or below a threshold. For example, for some tasks an error rate above 20 percent is unacceptable, whereas for others an error rate above 5 percent is unacceptable. The most straightforward analysis is to first establish an acceptable threshold for each task or each participant. Next, calculate whether that specific task's error rate or participant error count was above or below the threshold.
- In some situations you want to take into account that not all errors are created equal. Some errors are much more serious than others. It is possible to weight each type of error with a different value and then calculate an "error score"—for example, trivial, moderate, and serious. You could then weight each of those errors with a value of 1 (trivial), 2 (moderate), or 3 (serious). Then, simply add up the score for each participant using these weights. Divide all the scores by the number of participants for each task. This will produce an average "error score" for each task. The interpretation is a little different from an error rate. Essentially, you will be able to report that certain tasks have more frequent and/or serious errors than other tasks.

4.3.5 Issues to Consider When Using Error Metrics

Several important issues must be considered when looking at errors. First, make sure you are not double-counting errors. Double-counting happens when you assign more than one error to the same event. For example, assume you are counting errors in a password field. If a user typed an extra character in the password, you could count that as an "extra character" error, but you shouldn't also count it as an "incorrect character" error.

Sometimes you need to know more than just an error rate; you need to know *why* different errors are occurring. The best way to do this is by looking at each type of error. Basically, you want to try to code each error by type of error. Coding should be based on the various types of errors that occurred. Continuing with the password example, the types of errors might include "missing character," "transposed characters," "extra character," and so on. Or at a higher level, you might

have "navigation error," "selection error," "interpretation error," and so on. Once you have coded each error, you can run frequencies on the error type for each task to better understand exactly where the problems lie. This will also help improve the efficiency with which you collect the error data.

In some cases, an error is the same as failing to complete a task—for example, on a login page. If no errors occur while logging in, it is the same as task success. If an error occurs, it is the same as task failure. In this case, it might be easier to report errors as task failure. It's not so much a data issue as it is a presentation issue. It's important to make sure your audience clearly understands your metrics.

4.4 EFFICIENCY

Time-on-task is often used as a measure of efficiency, but another way to measure efficiency is to look at the amount of effort required to complete a task. This is typically done by measuring the number of actions or steps that participants took in performing each task. An action can take many forms, such as clicking a link on a web page, pressing a button on a microwave oven or a mobile phone, or flipping a switch on an aircraft. Each action a participant performs represents a certain amount of effort. The more actions taken by a participant, the more effort involved. In most products, the goal is to minimize the number of discrete actions required to complete a task, thereby minimizing the amount of effort.

What do we mean by effort? There are at least two types of effort: cognitive and physical. Cognitive effort involves finding the right place to perform an action (e.g., finding a link on a web page), deciding what action is necessary (should I click this link?), and interpreting the results of the action. Physical effort involves the physical activity required to take action, such as moving your mouse, inputting text on a keyboard, turning on a switch, and many others.

Efficiency metrics work well if you are concerned with not only the time it takes to complete a task but also the amount of cognitive and physical effort involved. For example, if you are designing an automobile navigation system, you need to make sure that it does not take much effort to interpret its navigation directions, since the driver's attention must be focused on the road. It would be important to minimize both the physical and cognitive effort to use the navigation system.

4.4.1 Collecting and Measuring Efficiency

There are five important points to keep in mind when collecting and measuring efficiency.

Identify the action(s) to be measured: For websites, mouse clicks or page views are common actions. For software, it might be mouse clicks or keystrokes. For appliances or consumer electronics, it could be button presses. Regardless of

the product being evaluated, you should have a clear idea of all the possible actions.

Define the start and end of an action: You need to know when an action begins and ends. Sometimes the action is very quick, such as a press of a button, but other actions can take much longer. An action may be more passive in nature, such as looking at a web page. Some actions have a very clear start and end, whereas other actions are less defined.

Count the actions: You must be able to count the actions. Actions must happen at a pace that can be identified visually or, if they are too fast, by an automated system. Try to avoid having to review hours of videotape to collect efficiency metrics.

Actions must be meaningful: Each action should represent an incremental increase in cognitive and/or physical effort. The more actions, the more effort. For example, each click of a mouse on a link is almost always an incremental increase in effort.

Look only at successful tasks: When measuring efficiency using the number of actions, you should only calculate it for successful tasks. It does not make sense to include task failures. For example, a participant may quit a task after only a few steps when he becomes hopelessly lost. If you used this data, it may look like he performed at the same level of efficiency as another participant who completed the task successfully with the minimum number of steps required.

Once you have identified the actions you want to capture, counting those actions is relatively simple. You can do it manually, such as by counting page views or presses of a button. This will work for fairly simple products, but in most cases it is not practical. Many times a participant is performing these actions at amazing speeds. There may be more than one action every second, so using automated data-collection tools is far preferable.

4.4.2 Analyzing and Presenting Efficiency Data

The most common way to analyze and present efficiency metrics is by looking at the number of actions each participant takes to complete a task. Simply calculate an average for each task (by participant) to see how many actions are taken. This analysis is helpful in identifying which tasks required the most amount of effort, and it works well when each task requires about the same number of actions. However, if some tasks are more complicated than others, it may be misleading. It's also important to represent the 95 percent confidence intervals (based on a continuous distribution) for this type of chart.

Shaikh, Baker, and Russell (2004) used an efficiency metric based on number of clicks to accomplish the same task on three different weight-loss sites: Atkins, Jenny Craig, and Weight Watchers. They found that users were significantly more

efficient (needed fewer clicks) with the Atkins site than with the Jenny Craig or Weight Watchers site.

Lostness

Another measure of efficiency sometimes used in studying behavior on the web is called "lostness" (Smith, 1996). Lostness is calculated using three values:

N: The number of *different* web pages visited while performing the task

S: The *total* number of pages visited while performing the task, counting revisits to the same page

R: The *minimum* (optimum) number of pages that must be visited to accomplish the task

Lostness, L, is then calculated using the following formula:

$$L = \text{sqrt}[(N/S - 1)^2 + (R/N - 1)^2]$$

Consider the example shown in Figure 4.9. In this case, the participant's task is to find something on Product Page C1. Starting on the homepage, the minimum number of page visits (R) to accomplish this task is three. On the other hand, Figure 4.10 illustrates the path a particular participant took in getting to that target item. This participant started down some incorrect paths before finally getting to the right place, visiting a total of six different pages (N), or a total of eight page visits (S). So for this example:

$$N = 6$$
$$S = 8$$
$$R = 3$$
$$L = \text{sqrt}[(6/8 - 1)^2 + (3/6 - 1)^2] = 0.56$$

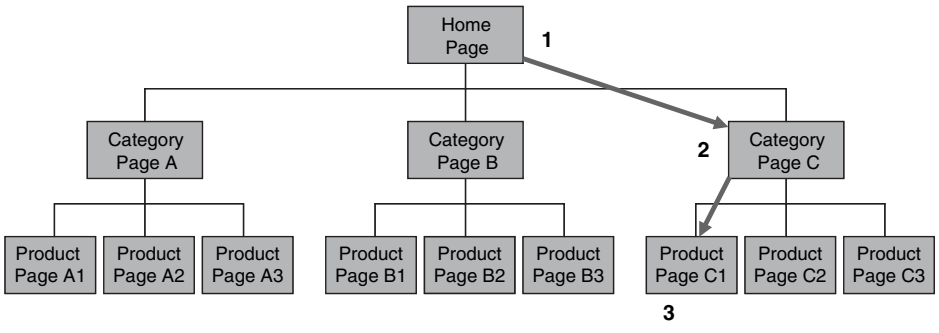


FIGURE 4.9

Optimum number of steps (three) to accomplish a task that involves finding a target item on Product Page C1 starting from the homepage.

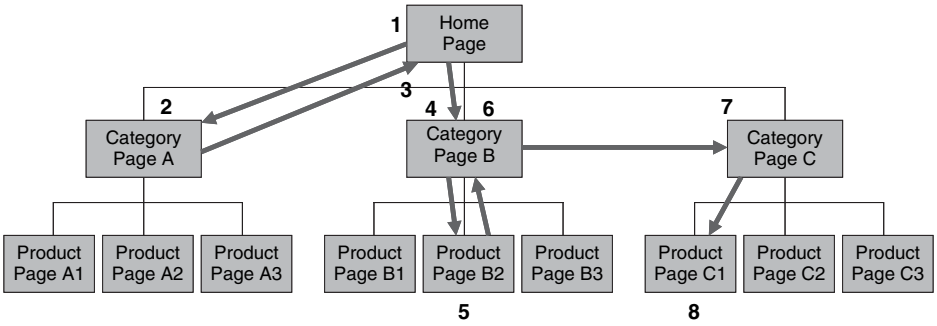


FIGURE 4.10

Actual number of steps a participant took in getting to the target item on Product Page C1. Note that each revisit to the same page is counted, giving a total of eight steps.

A perfect lostness score would be 0. Smith (1996) found that participants with a lostness score less than 0.4 did not exhibit any observable characteristics of being lost. On the other hand, participants with a lostness score greater than 0.5 definitely did appear to be lost.

Once you calculate a lostness value, you can easily calculate the average lostness value for each task. The number or percent of participants who exceed the ideal number of actions can also be indicative of the efficiency of the design. For example, you could show that 25 percent of the participants exceeded the ideal or minimum number of steps, and you could break it down even further by saying that 50 percent of the participants completed a task with the minimum number of actions.

4.4.3 Efficiency as a Combination of Task Success and Time

Another view of efficiency is that it's a combination of two of the metrics discussed in this chapter: task success and time-on-task. The Common Industry Format (CIF) for Usability Test Reports (NIST, 2001) specifies that the "core measure of efficiency" is the ratio of the task completion rate to the mean time per task. Basically, it expresses task success per unit time. Most commonly, time per task is expressed in minutes, but seconds could be appropriate if the tasks are very short, or even hours if they are unusually long. The unit of time used determines the scale of the results. Your goal is to choose a unit that yields a "reasonable" scale (i.e., one where most of the values fall between 1 and 100 percent).

Table 4.2 shows an example of calculating an efficiency metric, which is simply the ratio of the task completion to the task time in minutes. Of course, higher values of efficiency are better. In the example in the table, participants appear to have been more efficient when performing Tasks 5 and 6 than the other tasks. Figure 4.11 shows how this efficiency metric looks in a chart.

Table 4.2 Calculating an Efficiency Metric

Task	Completion Rate Percentage	Task Time (mins)	Percent Efficiency
1	65	1.5	43
2	67	1.4	48
3	40	2.1	19
4	74	1.7	44
5	85	1.2	71
6	90	1.4	64
7	49	2.1	23
8	33	1.3	25

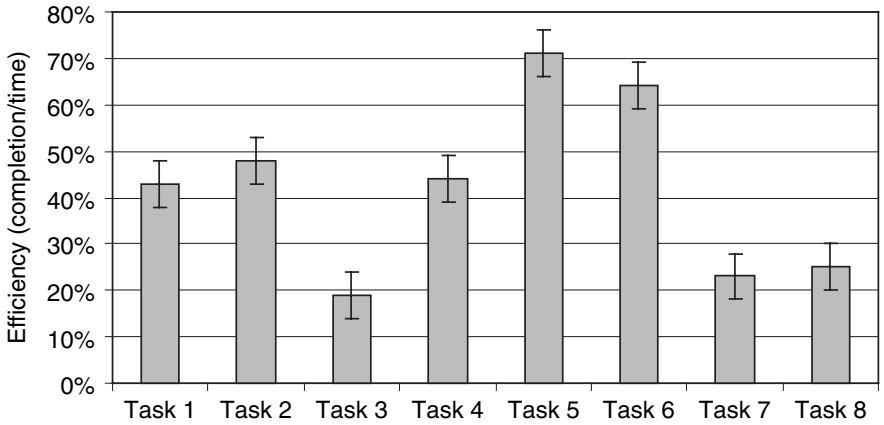


FIGURE 4.11

An example showing efficiency as a function of completion rate/time.

A slight variation on this approach to calculating efficiency is to count the number of tasks successfully completed by each participant and divide that by the total time spent by the participant on all the tasks (successful and unsuccessful). This gives you a very straightforward efficiency score for each participant: number of tasks successfully completed per minute (or whatever unit of time you used). If a participant completed ten tasks successfully in a total time of ten minutes, then that participant was successfully completing one task per minute overall. This works best when all participants attempted the same number of tasks and the tasks are relatively comparable in terms of their level of difficulty.

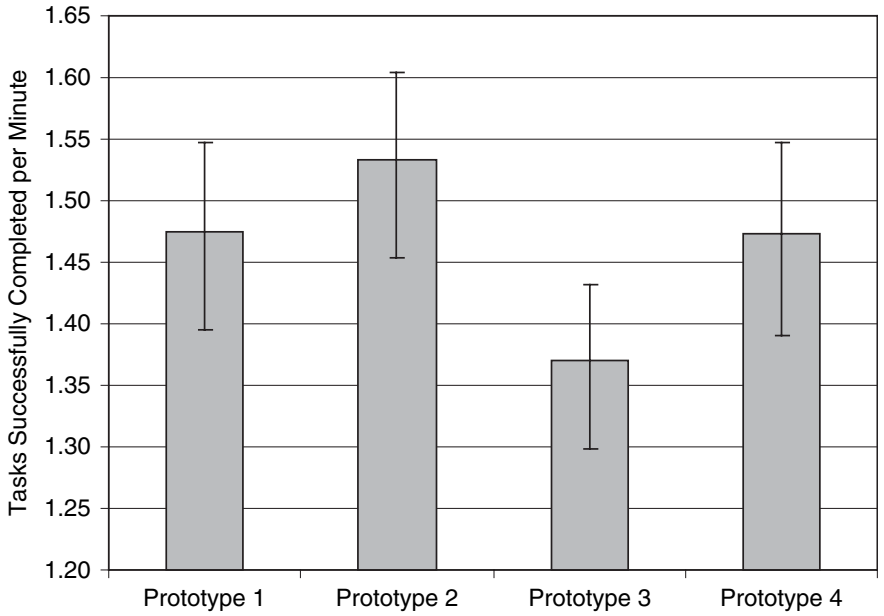


FIGURE 4.12

Average number of tasks successfully completed per minute in an online study of four different prototypes of navigation for a website. More than 200 participants attempted 20 tasks for each prototype. Participants using Prototype 2 were significantly more efficient (i.e., completed more tasks per minute) than those using Prototype 3.

Figure 4.12 shows the data from an online study comparing four different navigation prototypes for a website. This was a between-subjects study, in which each participant used only one of the prototypes, but all participants were asked to perform the same 20 tasks. More than 200 participants used each prototype. We were able to count the number of tasks successfully completed by each participant and divide that by the total time that participant spent. The averages of these (and the 95 percent confidence intervals) are shown in Figure 4.12.

4.5 LEARNABILITY

Most products, especially new ones, require some amount of learning. Usually learning does not happen in an instant but occurs over time as experience increases. Experience is based on the amount of time spent using a product and the variety of tasks performed. Learning is sometimes quick and painless, but it is at other times quite arduous and time consuming. Learnability is the extent to which something can be learned. It can be measured by looking at how much time and effort are required to become proficient with something. We believe that

learnability is an important usability metric that does not receive as much attention as it should. It's an essential metric if you need to know how someone develops proficiency with a product over time.

Consider the following example. Assume you're a usability specialist who has been asked to evaluate a time-keeping application for employees within the organization. You could go into the lab and test with ten participants, giving each one a set of core tasks. You might measure task success, time-on-task, errors, and even overall satisfaction. Using these metrics will allow you to get some sense of the usability of the application.

Although these metrics are useful, they can also be misleading. Because the use of a time-keeping application is not a one-time event, but happens with some degree of frequency, learnability is very important. What really matters is how much time and effort are required to become *proficient* using the time-keeping application. Yes, there may be some initial obstacles when first using the application, but what really matters is "getting up to speed." It's quite common in usability studies to only look at a participant's initial exposure to something, but sometimes it's more important to look at the amount of effort needed to become proficient.

Learning can happen over a short period of time or over longer periods of time. When learning happens over a short period of time, the participant tries out different strategies to complete the tasks. A short period of time might be several minutes, hours, or days. For example, if participants have to submit their timesheets every day using a time-keeping application, they try to quickly develop some type of mental model about how the application works. Memory is not a big factor in learnability; it is more about adapting strategies to maximize efficiency. Within a few hours or days, maximum efficiency is hopefully achieved.

Learning can also happen over a longer time period, such as weeks, months, or years. This is the case where there are significant gaps in time between each use. For example, if you only fill out an expense report every few months, learnability can be a significant challenge because you may have to relearn the application each time you use it. In this situation, memory is very important. The more time there is between experiences with the product, the greater the reliance on memory.

4.5.1 Collecting and Measuring Learnability Data

Collecting and measuring learnability data are basically the same as they are for the other performance metrics, but you're collecting the data at multiple times. Each instance of collecting the data is considered a trial. A trial might be every five minutes, every day, or once a month. The time between trials, or when you collect the data, is based on expected frequency of use.

The first decision is which types of metric you want to use. Learnability can be measured using almost any performance metric over time, but the most common

are those that focus on efficiency, such as time-on-task, errors, number of steps, or task success per minute. As learning occurs, you expect to see efficiency improve.

After you decide which metrics to use, you need to decide how much time to allow between trials. What do you do when learning occurs over a very long time? What if users interact with a product once every week, month, or even year? The ideal situation would be to bring the same participants into the lab every week, month, or even year. In many cases, this is not very practical. The developers and the business sponsors might not be very pleased if you told them the study will take three years to complete. A more realistic approach is to bring in the same participants over a much shorter time span and acknowledge the limitation in the data. Here are a few alternatives:

Trials within the same session. The participant performs the task, or set of tasks, one right after the other, with no breaks in between. This is very easy to administer, but it does not take into account significant memory loss.

Trials within the same session but with breaks between tasks. The break might be a distracter task or anything that might promote forgetting. This is fairly easy to administer, but it tends to make each session relatively long.

Trials between sessions. The participant performs the same tasks over multiple sessions, with at least one day in between. This may be the least practical, but most realistic, if the product is used sporadically over an extended period of time.

4.5.2 Analyzing and Presenting Learnability Data

The most common way to analyze and present learnability data is by examining a specific performance metric (such as time-on-task, number of steps, or number of errors) by trial for each task or aggregated across all tasks. This will show you how that performance metric changes as a function of experience, as illustrated in Figure 4.13. You could aggregate all the tasks together and represent them as a single line of data, or you could look at each task as separate lines of data. This can help to determine how the learnability of different tasks compares, but it can also make the chart harder to interpret.

The first aspect of the chart you should notice is the slope of the line(s). Ideally, the slope (sometimes called the learning curve) is fairly flat and low on the *y*-axis (in the case of errors, time-on-task, number of steps, or any other metric where a smaller number is better). If you want to determine whether a statistically significant difference between the learning curves (or slopes) exists, you need to perform an analysis of variance and see if there is a main effect of trial.

You should also notice the point of *asymptote*, or essentially where the line starts to flatten out. This is the point at which a participant has learned as much as she can and there is very little room for improvement. Project team members are

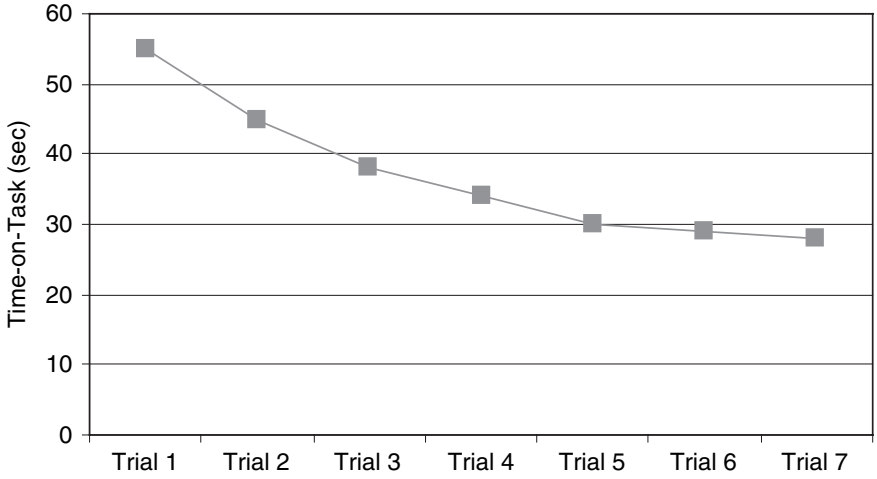


FIGURE 4.13

An example of how to present learnability data based on time-on-task.

always interested in how long it will take someone to reach maximum performance.

Finally, you should look at the difference between the highest and lowest values on the *y*-axis. This will tell you how much learning must occur to reach maximum performance. If the gap is small, users will be able to quickly learn the product. If the gap is large, users may take quite some time to become proficient with the product. One easy way to analyze the gap between the highest and lowest scores is by looking at the ratio of the two. Here is an example:

- If the average time on the first trial is 80 seconds and on the last trial is 60 seconds, the ratio shows that participants are initially taking 1.3 times longer.
- If the average number of errors on the first trial is 2.1 and on the last trial is 0.3, the ratio shows a 7 times improvement from the first trial to the last trial.

It may be helpful to look at how many trials are needed to reach maximum performance. This is a good way to characterize the amount of learning required to become proficient in using the product.

In some cases you might want to compare learnability across different conditions, as shown in Figure 4.14. In this study (Tullis, Mangan, & Rosenbaum, 2007), they were interested in how speed (efficiency) of entering a password changed over time using different types of on-screen keyboards. As you can see from the data, there is an improvement from the first trial to the second trial, but then the times flatten out pretty quickly. Also, all the on-screen keyboards were significantly slower than the control condition, which was a real keyboard.

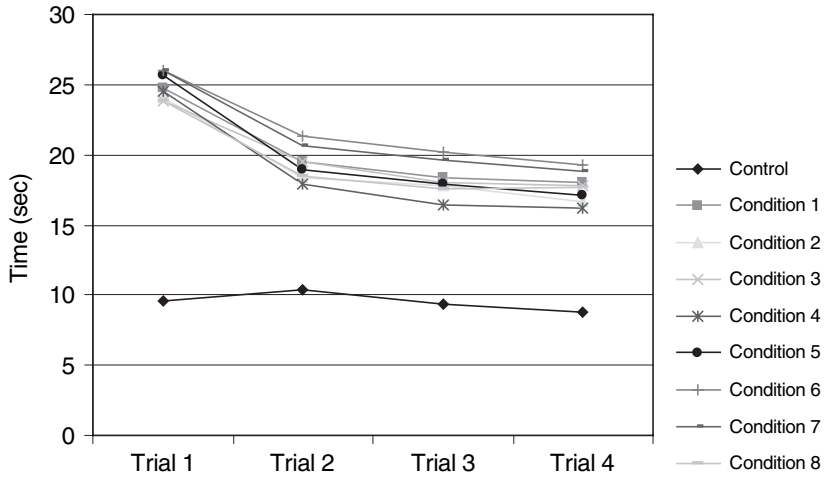


FIGURE 4.14

Looking at the learnability of different types of on-screen keyboards.

4.5.3 Issues to Consider When Measuring Learnability

Two of the key issues to address when measuring learnability are (1) what should be considered a trial and (2) how many trials to include.

What Is a Trial?

In some situations learning is continuous. This means that the user is interacting with the product fairly continuously without any significant breaks in time. Memory is much less a factor in this situation. Learning is more about developing and modifying different strategies to complete a set of tasks. The whole concept of trials does not make much sense for continuous learning. What do you do in this situation? One approach is to take measurements at specified time intervals. For example, you may need to take measurements every 5 minutes, every 15 minutes, or every hour.

In one usability study we conducted, we wanted to evaluate the learnability of a new suite of applications that would be used many times every day. We started by bringing the participants into the lab for their first exposure to the applications and their initial tasks. They then went back to their regular jobs and began using the applications to do their normal work. We brought them back into the lab one month later and had them perform basically the same tasks again (with minor changes in details) while we took the same performance measures. Finally, we brought them back one more time after another month and repeated the procedure. In this way, we were able to look at learnability over a two-month period.

Number of Trials

How many trials do you need? Obviously there must be at least two, but in most cases there should be at least three or four. Sometimes it's difficult to predict

where in the sequence of trials the most learning will take place, or even *if* it will take place. In this situation, you should err on the side of more trials than you think you might need to reach stable performance.

4.6 SUMMARY

Performance metrics are powerful tools to evaluate the usability of any product. They are the cornerstone of usability and can inform key decisions, such as whether a new product is ready to launch. Performance metrics are always based on participants' behavior rather than what they say. There are five general types of performance metrics.

1. *Task success* metrics are used when you are interested in whether participants are able to complete tasks using the product. Sometimes you might only be interested in whether a user is successful or not based on a strict set of criteria (binary success). Other times you might be interested in defining different levels of success based on the degree of completion, the experience in finding an answer, or the quality of the answer given.
2. *Time-on-task* is helpful when you are concerned about how quickly users can perform tasks with the product. You might look at the time it takes all participants to complete a task, a subset of participants, or the proportion of participants who can complete a task within a desired time limit.
3. *Errors* are a useful measure based on the number of mistakes made while attempting to complete a task. A task might have a single error opportunity or multiple error opportunities, and some types of errors may be more important than others.
4. *Efficiency* is a way of evaluating the amount of effort (cognitive and physical) required to complete a task. Efficiency is often measured by the number of steps or actions required to complete a task or by the ratio of the task success rate to the average time per task.
5. *Learnability* involves looking at how any efficiency metric changes over time. Learnability is useful if you want to examine how and when participants reach proficiency in using a product.