CHAPTER 5

Develop the Test Plan

The test plan is the foundation for the entire test. It addresses the how, when, where, who, why, and what of your usability test. Under the sometimes unrelenting time pressure of project deadlines, there could be a tendency to forgo writing a detailed test plan. Perhaps, feeling that you have a good idea of what you would like to test in your head, you decide not to bother writing it down. This informal approach is a mistake, and it invariably will come back to haunt you.

Why Create a Test Plan?

A sound approach is to start writing the test plan as soon as you know you will be testing. Then, as the project proceeds, continue to refine it, get feedback, buy-in, and so forth. Of course, there is a limit to flexibility, so prior to the test you need to set a reasonable deadline after which the test plan may not change. Let that date also serve as the point at which the product can no longer change until after the test. You may find that the test plan is the only concrete milestone at that point in time in the development cycle and, as such, serves an important function.

Once you reach the cutoff date, do all that you can to freeze the design of the product you will be testing. Additional revisions may invalidate the test design you have chosen, the questions you ask, even the way you collect data. If you are pressured to revise the test after the cutoff date, make sure that everyone understands the risks involved. The test may be invalidated, and the product may not work properly with changes made so close to the test date.

The following are some important reasons why it is necessary to develop a comprehensive test plan, as well as some ways to use it as a communication vehicle among the development team.

It Serves as a Blueprint for the Test

Much as the blueprint for a house describes exactly what you will build, the test plan describes exactly how you will go about testing your product. Just as you don't want your building contractor to ''wing it'' when building your house, so the exact same logic applies here. The test plan sets the stage for all that will follow. You do not want to have any loose ends just as you are about to test your first participant.

It Serves as the Main Communication Vehicle

The test plan serves as the main communication vehicle among the main designer and developer, the test moderator, and the rest of the team. The test plan is the document that all involved members of the development team, as well as management (if it is interested and involved), should review in order to understand how the test will proceed and to see whether their particular needs are being met. You use it to get buy-in and feedback from other members to ensure that everyone agrees on what will transpire. Because projects are dynamic and change from day to day and from week to week, you do not want someone to say at the end of the test that his or her particular agenda was not addressed. Especially when your organization is first starting to test, everyone who is directly affected by the test results should review the test plan. This makes good business sense and political sense as well.

It Defines or Implies Required Resources

The test plan describes or implies required resources, both internal and external. Once you delineate exactly what will happen and when, it is a much easier task to foretell what you will need to accomplish with your test. Either directly or by implication, the test plan should communicate the resources that are required to complete the test successfully.

It Provides a Focal Point for the Test and a Milestone

Without the test plan, details get fuzzy and ambiguous, especially under time pressure. The test plan forces you to approach the job of testing systematically, and it reminds the development team of the impending dates. Having said all that, it is perfectly acceptable, and highly probable, that the test plan will be developed in stages as you gradually understand more of the test objectives and talk to the people who will be involved. Projects are dynamic, and the best

laid plans will change as you begin to approach testing. By developing the test plan in stages, you can accommodate changes. For example, as your time and resource constraints become clearer, your test may become less ambitious and simpler. Or, perhaps you cannot acquire as many qualified participants as you thought. Perhaps not all modules or sections of the document will be ready in time. Perhaps your test objectives are too imprecise and need to be simplified and focused. These are all real-world examples that force you to revise the test and the test plan.

NOTE **Remember to keep the end user in mind as you develop the test plan. If you are very close to the project, there is a tendency to forget that you are not testing the product, you are testing its relationship to a human being with certain specific characteristics.**

The Parts of a Test Plan

Test plan formats will vary according to the type of test and the degree of formality required in your organization. However, the following are the typical sections to include, along with a description of each one. At the end of this chapter is a sample test plan.

- Purpose, goals, and objectives of the test
- Research questions
- Participant characteristics
- Method (test design)
- Task list
- Test environment, equipment, and logistics
- Test moderator role
- Data to be collected and evaluation measures
- Report contents and presentation

These parts are discussed in detail in the following sections, with the exception of the role of the test moderator, which gets its own chapter, Chapter 4, because it merits a special discussion.

Review the Purpose and Goals of the Test

For this part of the document, you need to describe at a high level the reasons for performing this test at this time. You need not provide the very specific objectives or problems to be explored here — rather, the major focus

or impetus is the key point, often from the viewpoint of your organization. For example:

- Is the test attempting to resolve problems that have been reported by the company's call center or support desk?
- Have server logs or web usage statistics shown that visitors to your company's web site leave the site at a particular point in a process that leaves a transaction incomplete?
- Has a new policy recently been instituted stating that all products must be tested before release?
- Does management feel it is critical for the development team to see real users at this time?

It is okay if the test purpose remains at a high level, because the research questions and problem statements will reduce the goal(s) to measurable statements. The important point is that the testing be tied to business goals within the organization and that testing is the most appropriate technique for addressing the problem or opportunity.

When Not to Test

Following are some rather vague, inappropriate reasons for usability testing a product. These are rarely placed on paper but are usually communicated via word of mouth. They are *not* sound reasons for testing, and invariably they often come back to sabotage the project.

- You can improve the user experience (you may be able to test one part of the customer experience but not all the touch points your company has with customers).
- Everyone else has a usability testing program (everyone else has many things).
- The meeting rooms used for testing are available the third week of the month (so is the cafeteria every evening).
- Lou just went to the latest ACM SIGCHI (Association for Computing Machinery Special Interest Group on Computer-Human Interaction) conference and learned about this really neat testing technique (let Lou promote the technique's benefits to the organization first).
- You want to see if there is a need for this type of product in the marketplace (backwards logic; a focus group or survey is a more appropriate technique early on).

You might say to yourself, especially if you are eager to begin usability testing, ''As long as we test, I don't care what the reasons are. We'll worry about the consequences later.'' And for the short term, there is no problem with any of the reasons stated previously. However, in the long term, if

you want testing to become an integral part of the way your organization develops products, you must tie testing to the needs of the product and to the organization's overall business needs. Otherwise, you run the risk of your testing becoming one more fad, one more of the latest approaches that come and go with the seasons.

Good Reasons to Test

The following list gives some more rational reasons for conducting a test, which should result in successful outcomes and pave the way for future tests.

- You want to understand whether both of your major types of users can use the product equally well.
- You want to know whether or not the documentation is able to compensate for some acknowledged problems with the interface.
- You have received numerous complaints associated with using the product. You are interested in determining the exact nature of the problem and how you will fix it within your development budget for this year.

Figure 5-1 shows one example of purpose and goals for a usability test of a hotel reservations web site.

Communicate Research Questions

This section is the single most important one in the test plan, because it describes the issues and questions that need to be resolved and focuses the research, as well as the rest of the activities associated with planning, designing, and conducting the test. It is essential that the research questions be as precise, accurate, clear, and measurable (or observable) as possible. Even when conducting exploratory testing in the early stages of developing a product, which is typically less structured, you still need to accurately describe what you hope to learn.

Without a clear succinct research question(s), you might find yourself in the unenviable position of conducting a wonderful test that neglects to answer the

Overall objectives for the study

We will gather baseline data about the overall effectiveness of H.com. The goals of this study are to:

- Assess the overall effectiveness of www.H.com for different types of users performing basic, common tasks.
- Identify obstacles to completing room reservations on the site.
- Create a repeatable usability study protocol.

Figure 5-1 Sample purpose and goals for a usability test

key concern of developers on the project team. Or, you might find yourself with a test whose development bogs down in controversy because no one can agree on what to test. Speaking from experience, we have seen test preparations move in circles and the test itself result in controversy because the test objectives were never committed to paper.

The following are two examples of unfocused and vague research questions.

- **EXAMPLE 1.** Is the current product usable?
- **EXAMPLE 2.** Is the product ready for release or does it need more work?

The difficulty with these questions is *not* that they do not make sense. Rather, they are incomplete and vague. They neither state nor imply how to measure or quantify the results. A test based on these statements will invariably bias the results favorably. Why? If those involved cannot agree on what problems or issues need to be resolved, how do you know when you have found one? Of course, in those circumstances, the tendency will be *not* to find any problems.

The table below shows an example of several more appropriately focused research questions for several types of products. The research question(s) should originate with discussions with the development team or with individual developers, technical writers, marketing personnel, and so on. Do not be surprised if they have difficulty in pinning down the test objectives and if they can communicate only the most general questions or objectives. This may be an indication that:

- They are not quite ready to test.
- They need a greater understanding and education of the goals, intent, and process of testing.
- They need help in formulating their objectives into research questions that can be measured or observed. Do not be afraid to jump in and help.

If you find that you are having unusual difficulty designing the test and/or appropriate measures, or deciding on the appropriate end users, or even designing the data collection form, you might return to the research questions to see if they are clear or need further clarification.

PRODUCT	RESEARCH QUESTIONS
Web sites	How easily do users understand what is clickable? How easily and successfully do users find the products or information they are looking for? How easily and successfully do users register for the site? Where in the site do users go to find Search? Why? How easily can users return to the home page?

(*continued*)

PRODUCT	RESEARCH QUESTIONS
Small interfaces	How easily do users switch between modes on multi-purpose buttons? How well do users understand the symbols and icons? Which ones are problematic? Why? How easily do users download updates and features? How quickly can users perform common tasks?
Hardware	How easily and successfully can users use all buttons on the control panel? Can users use the control panel without assistance or training? How easily can users find the correct input and output ports? How easily can users change settings in the menus?
Online and written documentation	Do users go to online help when they encounter error messages? How easily do users find topics they are looking for in the online help? How well do the topic titles reflect what users are looking for? How well do they understand the content of the topics they find? How helpful is the topic content? Which parts of each topic do users pay attention to? Can users easily switch between reading the online help and interacting with the interface to complete the task?
Software	How closely does the flow of the software reflect how the user thinks of the work flow? How easily and successfully do users find the tools or options they want? Do users use the toolbar icons or the standard menus? Why? Is the response time a cause of user frustration or errors?
General	What obstacles prevent users from completing installation and set up? Can users perform common tasks within established benchmarks? What are the major usability flaws that prevent users from completing the most common tasks? How does ease-of-use compare in the planned release to the last release? How does ease-of-use compare between our product and the competition? Is there an appropriate balance of ease of use and ease of learning?

Research questions

In addition, in this study will try to answer these questions:

- How easily and successfully do travelers get started with making a reservation on the site?
- Does the starting point make any difference in whether travelers are successful in reaching their goal on H.com? If so, what are the differences?
- What paths do travelers take to completing a booking?
- How well does the site support the paths and goals of the travelers? That is, how closely does the organization and flow of the site match travelers' expectations?
- What obstacles do travelers encounter on the way to completing a booking, whether using a credit card or rewards?
- What questions do travelers ask as they work through their reservation?
- How do travelers feel about how long it takes them to complete an online booking, both the perceived of time and the number of steps?

Figure 5-2 Sample research questions

Figure 5-2 shows an example of research questions from a usability test of a hotel reservations web site.

Summarize Participant Characteristics

This section of the test plan describes the characteristics of the end user(s) of the product/document that you will be testing. It is important to work closely with others in your organization to determine the characteristics of the target users. For detailed procedures on how to establish the user profile and acquire participants, see Chapter 7. A basic example of participant characteristics for a usability test of a hotel reservations web site appears in Figure 5-3.

One thing to remember when describing the participant characteristics is to use the right number of participants. When it comes to selecting the number of participants to employ for a test, the overriding guideline is ''You cannot have too many participants.'' When thinking about achieving statistically valid results, small sample sizes lack the statistical power to identify significant differences between groups. For a true experimental design, you must use a minimum of 10 to 12 participants per condition. However, for the purpose of conducting a less formal usability test, research has shown that four to five participants who represent one audience cell will expose about 80 percent of the usability deficiencies of a product for that audience, and that this 80 percent will represent most of the major problems. Of course, if you have the time and resources to study more than four or five participants, by all means do so. It is possible that the additional 20 percent of deficiencies you might find could be important for your product.

We have conducted many tests that held true to the preceding principle. In one, Jeff tested eight participants and discovered about 80 percent of the problems within the first four participants. However, participant 8, the last one, performed a particularly grievous error on one task that would have required a service call for the product. This would never have been uncovered

Characteristic	Desired number of participants
Participant type	
pilot	1
regular	12
backup	2
Total number of participants	**14**
Travel frequency	
infrequently: 1–5 trips per year	4
moderately often: 6–12 trips per year	4
very often: 13 or more trips per year	4
Types of travel	
mostly business	6
mostly leisure	6
Booking experience	
book their own trips and accommodations	all
book online most of the time	6
book on the phone or other method	6
Age	
21–30	2–3
31–40	4–5
41–50	4–5
51–60	2–3
Gender	
female	6
male	6

Figure 5-3 Sample participant characteristics and desired mix

had we only tested four participants. Until you become experienced at testing, employing more participants decreases the probability you will miss an important problem, while providing additional opportunities to practice your moderating skills.

If you find you have very limited time and budget, you may want to institute a practice of ''discount'' usability testing, in which you would run several small, iterated usability tests over time. That is, conduct a test with 4 or 5 participants from one cell and one or two conditions, incorporate the findings into the interface, and then conduct another test with a similar set of participants and conditions. Over three or four tests, you end up with a large sample of participants, but the development team is able to accommodate changes in between tests.

Describe the Method

This section of the test plan is a detailed description of how you are going to carry out the research with the participants, and how the test session will

unfold. Essentially, it is a synopsis of your test design. It should provide an overview of each facet of the test from the time the participants arrive until the time they leave, in enough detail so that someone observing the test will know roughly what to expect. If you are questioning why this amount of detail is necessary in the test plan, the following reasons should satisfy your curiosity.

- It enables others to understand and visualize what will happen so that they can comment and make suggestions accordingly.
- It enables you as the test developer to focus on what has to be done and the types of materials that have to be developed before participants arrive.
- It reveals the need to communicate your plans to additional resources whom you might have forgotten, such as a receptionist who will greet the participants in a corporate lobby when they first arrive.
- It allows multiple test moderators (if that is required by the test design) to conduct the test in as similar a manner to each other as possible.

Test design is one of the more highly specialized skills required of a usability professional, often requiring knowledge of experimental design and method and basic statistical analysis. Designing a test requires one to clearly identify and understand the test objectives, and then to select the test design that will effectively ferret out the answers to the questions posed. If the test design is flawed or if the test is carried out with little attention to experimental rigor, then the results will be suspect. Not only can this result in faulty recommendations, but it also sabotages the progress of usability engineering per se within the organization. Therefore, the first few times that you conduct a usability test, get advice and feedback on your test design from someone more experienced than you.

The test design is mainly predicated upon your test objectives — what you need to learn about the product and its audience. The design will be greatly affected by your resources, your constraints, and your creativity. Constraints are time, money, management backing, development team support, ability to acquire participants, and other real-world concerns. The following sections give examples of test designs for some of the most common situations you will face. Following that, we present some guidelines for ensuring experimental rigor.

The simplest test design, shown in the table in the next section, consists of testing several different users, all from one type of user group (e.g., older adults), and having them perform a series of representative tasks on different parts of the web site.

Independent Groups Design or Between Subjects Design

This is called an independent groups design because each part of the web site is tested by a unique set of users. For our example, shown in the table below, this design requires 15 participants and mitigates the potential transfer of learning effects caused by doing one set of tasks prior to performing other similar tasks. In other words, performing Task A may help one to perform Task B, and mask any usability problems associated with Task B. You can also use this design if the tasks are extremely lengthy and there is a possibility that the participants may become fatigued.

TASK A SIGN UP TO BECOME A MEMBER	TASK B FIND ONLINE CLASSES TO TAKE	TASK C FIND VOLUNTEER OPPORTUNITIES
Terry	Pat	Tracy
Lesley	Michael	Dana
Lisa	Andrea	Duane
Kim	Erin	Aaron
Blair	Paula	Janet

Within-Subjects Design

Perhaps testing 15 participants is simply out of the question. Instead of 15, you could get by with only five participants by having each one perform all three modules as shown in the table below. This is called a *within-subjects* design. However, you have the same problem of transfer of learning effects to consider. To mitigate these effects, you must use a technique called *counterbalancing*, whereby the order of tasks is either randomized or balanced out. By varying the order of the presentation of tasks, you can limit the effects of learning transfer.

TASK A SIGN UP TO BECOME A MEMBER	TASK B FIND ONLINE CLASSES TO TAKE	TASK C FIND VOLUNTEER OPPORTUNITIES
Terry	Terry	Terry
Lesley	Lesley	Lesley
Lisa	Lisa	Lisa
Kim	Kim	Kim
Blair	Blair	Blair

To counterbalance, you vary the presentation order of modules as shown in the table below, with each participant performing modules in a different order. By randomizing the order of the modules, you minimize the transfer effects while requiring only four participants. However, there are still some issues to resolve. If the order of modules would normally be sequential in real life (e.g., the modules required to set up a piece of hardware), then you have an important decision. Is it more critical to provide a realistic task order for users and possibly mask some usability problems on later tasks (possibly measuring whether participants learn as they progress through the modules), or is it more crucial to provide a random order of tasks (which is possible in the lab) and risk confusing and alienating the participant? Most would argue that you should retain the sequential order. If you decide to do so, you will still need to address possible transfer effects, possibly by using prerequisite training to equalize participants' experience before performing. In addition, you may need to conduct each session with breaks to allow participants to rest.

PARTICIPANT	TASK SEQUENCE
Terry	A, B, C
Lesley	B, C, A
Lisa	C, A, B
Kim	B, A, C
Blair	C, B, A

Testing Multiple Product Versions

Now let's look at another common situation. Suppose that you want to compare two different versions of a product, Version A and Version B, to see which one shows more promise as your ultimate design. (These are known as different ''conditions.'') Additionally, you want to see whether performance varies for either of two user groups, call them supervisors and technicians. This will result in a 2 × 2 matrix design as shown in the following table.

GROUP	VERSION A	VERSION B
Supervisors	4	4
Technicians	4	4

If you use an *independent groups* design whereby each cell in the table you use to describe your test design is populated by a different set of participants, then this design will require 16 participants to satisfy the four different conditions: Four supervisors will use Version A and four technicians will use Version A, and so on. Suppose though that you only want to use eight participants. You could simply populate each cell with only two participants, but that is increasing the risk that the data for any one group will be meaningless. Instead,

let each person in the two groups, supervisors and technicians, try each of the versions, one after the other, as shown in the table below. As with the previous example, there may be an unfair advantage for the version that is tested last, because the participant may learn to perform the tasks while using the first version. On the other hand, it may even reverse the effect; the participant may learn the first version and have difficulty adapting to the second version because it is so different. In either case, your results may be biased.

SUPERVISORS	VERSION	TECHNICIANS	VERSION
Ginny	A, B	Laurie	A, B
Stephanie	B, A	Janice	B, A
Ken	A, B	Arnold	A, B
James	B, A	Andrew	B, A

To account for these potential differences, you will again counterbalance the order of presentation of the versions. As shown in the table above, for eight participants, some participants will do Version A first, and others will do Version B first. Note that each version is performed as many times in the first position as it is in the last position, which negates the potential biasing effects.

Testing Multiple User Groups

Now let's look at a slightly more complex, yet realistic scenario. Suppose your user profile consists of two different user groups, managers and clerks, who will be using your product. One of your test objectives is to see if there are differences in ability to use the product between or among user groups. In addition, you also want to see if there are differences in novice and experienced users within each group. You will therefore need to vary experience and job type, each of which will have two levels. Once again, you will use a matrix design, as shown below.

GROUP	NOVICE	EXPERIENCED
Managers	4	4
Clerks	4	4

Each one of the four conditions or cells shown in the table above will be populated with a different set of participants. If you want to acquire at least four participants per cell, as shown, you will need a total of 16 participants. If this is too many participants for your budget and time, (four participants is about the bare minimum per group required to evaluate group differences), then you *cannot* simply apply a within-subjects design. Instead, you will either

have to limit each cell to fewer participants or simplify the study. Remember, limiting a cell to less than four participants severely limits the conclusions you can draw about each group. You will probably need to simplify the research to exclude a study of group differences (see Figure 5-4).

Methodology

This usability study will be somewhat exploratory but will also gather assessment data about the effectiveness of `www.H.com`. Participants will fall into three groups by the starting point they use to perform the main task, which is to reserve a room. We will collect data about error and success rates as well as qualitative data about participants' experiences using the site.

We will use a between-subjects design

In this between-subjects study, each participant will work through one task path (in a within-subjects study, each participant would try all paths in counterbalanced order). I will conduct up to 30 individual 45-minute usability study sessions. Each participant will perform one of three major task "paths" using `www.H.com`. I'll use 15 minutes of each session to explain the session to the participant, review basic background information with the participant, and then conduct a post-test debriefing interview. During the middle 30 minutes of the session, participants will work to reserve a room at an H property in a major U.S. city.

Session outline and timing

The test sessions will be 45 minutes long. I will use 15 minutes of each session for pre-test introductions and post-test debriefing interviews. The sessions will take place at Shugoll Research in Bethesda.

Pre-test arrangements

Have the participant:

- Review and sign nondisclosures and recording permissions.
- Fill out a background questionnaire (with the same questions as the screener).

Introduction to the session (2 minutes)

Discuss:

- Participant's experience with usability studies and focus groups.
- Importance of their involvement in the study.
- Moderator's role.
- Room configuration, recording systems, observers, etc.
- The protocol for the rest of the session.
- Thinking aloud.

Background interview (3 minutes)

Discuss the participant's:

- Experiences booking their own travel.
- Reasons for booking their own travel.

Tasks (30 minutes)

Participants will start at one of three points to reserve a room at an H hotel in a major U.S. city where H has multiple properties.

Post-test debriefing (10 minutes)

- Ask broad questions to collect preference and other qualitative data.
- Follow up on any particular problems that came up for the participant.

Figure 5-4 High-level description of a test method

NOTE **If you are new to the game and are not confident that you can conduct a test with experimental rigor, then by all means keep the test simple. The more straightforward the test, the easier it is to keep everything consistent from session to session. It is better to attain meaningful results from a smaller, simpler study than to acquire a wealth of meaningless data from a larger study. Do some usability testing as early and often as possible. It need not be elaborate to be useful or cost-effective.**

List the Tasks

The task list comprises those tasks that the participants will perform during the test. The list should consist of tasks that will ordinarily be performed during the course of using the product, documentation, and so on.

There are two stages to developing these tasks. In the early stages of developing the test, the task list description is intended only for members of the project team and not for eventual participants. You need to supply only enough detail so that reviewers of the test plan can judge whether the tasks are the correct ones and are being exercised properly.

Later, you will expand the tasks into full-blown task scenarios, which are presented to the participants. The scenarios will provide the realistic details and context that enable the participants to perform tasks with little intervention from the test moderator. Expanding the initial tasks into task scenarios is covered in Chapter 8. For now, your task list need only include the following.

Parts of a Task for the Test Plan

For the test plan, you need only touch on four main components of each task:

A brief Description of the Task

Include only enough detail at this time to communicate the task to the project team. A one-line description is usually enough.

The Materials and Machine States Required to Perform the Task

Context is everything in usability testing. As the test moderator, you may actually be providing these materials or simulating the machine states if the product is in an early stage. For example, if you were testing a web site before the screens are coded or prototyped, you might provide printed wire-frame drawings of the pages. Or if the page were available in a file on the

computer but not hooked up as part of a working prototype yet, you (or the participant) might open that file on the screen for viewing at the appropriate time.

Or, perhaps parts of the test will be performed with documentation, while other sections will not. For example, if you are testing how well instructions work for installing a wireless network, and the later tasks will be done without documentation, such as specifying drive designations on the new network, you need to specify this. If it is appropriate and helpful, your task list might also include components of the product that are being exercised for that particular task. If, for example, a task asks a participant to enter a customer name into an online form, you might specify the screens or web pages that the participant will navigate during task completion. This helps to give you a sense of whether the full system is being exercised or not.

A Description of Successful Completion of the Task

How will you measure success? It is amazing how much disagreement there will be over this question and how often developers have differing opinions on what represents successful completion of a task. When you include *successful completion criteria* (SCC) with the task description, you add precision to what you are measuring and how you view the task. SCC define the boundaries of your task and help to clarify test scoring. When you have difficulty ascertaining the SCC, it reflects the development team's confusion about the product design. Establishing and documenting the SCC is a good exercise just for that reason alone.

Criteria for successful completion can include reaching a certain point in the task or screen flow, a maximum number of errors or wrong turns (for information-finding tasks), and whether you will consider the task ''complete'' if the participant reaches the appropriate end point but makes mistakes along the way.

Timing or Other Benchmarks

You may want to use time as a criterion for success or as a benchmark. If you do set benchmarks that are based on timing, it's recommended you do this very thoughtfully and under just the right circumstances. For example, time-on-task is a good measure for validation/summative tests, but it is rarely appropriate for early exploratory or formative tests. It is inadvisable to measure time-on-task if you're also asking participants to think aloud, because doing so typically slows task performance. For more about benchmark timings, see the Benchmark Timings sidebar. If you don't want to use time as a benchmark, you could use error rates; for example, completing a task with no errors of any kind.

ABOUT BENCHMARK TIMINGS THAT ESTABLISH THE MAXIMUM TIME LIMITS FOR PERFORMING

If appropriate, establish *benchmarks* that represent either the average or maximum time to perform the task. Benchmarks help to evaluate participant performance during a test. While they are not absolutely necessary, they can help you to monitor and evaluate the results of a test session more precisely, because successful participant performance is a reflection of both correct behavior and timely completion.

For example, if a participant takes 5 minutes to correctly enter his or her name and address on an email system, the design is obviously flawed from almost anyone's standards, and you need to know that. During the test, you need to track when a participant is outside the boundaries of some designated maximum time. You may choose to intervene at that point, or let the participant continue and note on your data collection form that he or she "maxed out." You will certainly have to stop the participant eventually, if he or she cannot complete the task correctly and to continue to collect data on other areas of the product.

It is important to determine and arrive at fair and reasonable benchmarks. There are a number of ways to do this, but before describing them, it should be emphasized that they need not be deadly accurate. In fact, if you are conducting iterative, ongoing testing, you will be revising the benchmarks from test to test as you learn more about realistic time frames for task completion. Sources of benchmark times include:

- **Any original case studies, interviews, or customer visits you may have performed or been privy to. You should not only note what tasks the end users perform but also how long they typically take. Obviously, not only is task definition essential for testing, but it should also be an integral part of the design process. (Better late than never if you are the first to ascertain the tasks that your end users will perform.)**
- **Any usability objectives that were included as part of the product or functional specification. Typically, the usability objectives include targets for time to complete functions or tasks.**
- **Any usability data from previous tests that were performed.**
- **Polling in-house end users who fit the user profile in one's own company. Simply asking them how long it takes them to perform common tasks will get you started.**

Because time benchmarks are subjective, they may be controversial. Product developers may rightfully feel that the benchmarks should be longer than the test moderator provides. To anticipate this potential controversy before the test begins, it pays to give developers the benefit of the doubt by erring on the side of overly generous benchmarks.

(continued)

ABOUT BENCHMARK TIMINGS THAT ESTABLISH THE MAXIMUM TIME LIMITS FOR PERFORMING *(continued)*

Jeff established benchmarks for one test for an organization with no previous usability testing experience. The text was for a hardware product that would be tested with documentation. Jeff had three engineers provide estimates of the maximum time that they felt a user would need to correctly perform each task on the test. He also had three technical writers on the project give the estimates because their perspective on the end user was different. He then averaged all estimates, and, to give everyone the benefit of the doubt, he multiplied the average for each task by a constant of 2.5 to come up with the maximum time for a participant to complete the task. This constant was rather arbitrary and quite generous. Jeff simply wanted everyone to feel that the participants were given ample time before the task was classified as "incomplete." The generosity was due to Jeff's confidence given his familiarity with the product design and its potential flaws as well as participants exposing the problematic areas, even with the generous time allotments.

As it turned out, some of the tasks took up to three times longer than even these generous benchmarks, which really drove home the point about difficulties. Experience has taught the authors that poor product design will make itself known eventually.

Measuring time on tasks is not always the best, most accurate measure of task success. If you are asking participants to think out loud, doing so takes time and unnaturally lengthens the duration. Instead, you may want to count only errors against the success criteria or completion criteria along with numbers and types of prompting.

Tips for Developing the Task List

While this may seem straightforward, it is a very subtle process. The trick is to *indirectly* expose usability flaws by having the participants perform tasks that use the parts of the product in question. What you are really testing is the *relationship* of your product to the end user. From the end user's viewpoint, your product and its associated documentation are a means to an end, either used to solve a problem or provide a service.

The tasks that you develop for the test need to reflect this relationship and, as much as possible, allow the test to expose the points at which the product becomes a hindrance rather than a help for performing a task. Let's look at a simple example of a task to satisfy a test objective and, in so doing, review some possible pitfalls.

Example Task: Navigation Tab on a Web Site

Suppose that one of your test objectives is to test how easy it is to understand a label for a tab that appears on an image-sharing web site that amateur and professional photographers use. The test objective is written as, ''Establish whether users can understand the meaning of the XYZ label.'' There are six tabs with text labels on the web site, but the XYZ label is the problematic one. It's called Organize.

On the current version of the web site, users expect to use the feature to change the order in which their images appear on the viewing pages, but this feature is for organizing images into categories.

If you simply take the objective at face value (''Establish whether users can understand the meaning of the XYZ label.''), you might decide to have a task that has the test moderator:

Show the participants the XYZ label and have them explain its meaning to you.

In other words, the test moderator will get feedback about the label. This seems simple and direct, because the label is the offending aspect of the product. However, this is oversimplifying the situation. By performing a simple analysis, you ascertain that there are actually three discrete processes associated with correctly using the simple label.

1. Noticing the label
2. Reading the label
3. Processing the information and responding correctly

In addition, these three processes occur within the very specific context of using the web site to post images on the web:

- If you simply show the participants the label, you only address the second and third processes. You will not know if the participants even notice the label, which precedes the other behaviors. You will also negate the entire context. In the course of using the web site, the participants will perform a particular task(s) at the time when they are supposed to be reading the label, not having someone point out the label and ask them what they think. This ''context'' is critical because it dramatically affects their ability to process information.
- You also need to address how the location of the label on the web page affects things. If it resides among five other labels and other actions, you should see how the participants perform with those potential distractions in place.

Tasks

Participants start from one of three starting points: All participants will use `www.H.com` to book a hotel room (up to the point of entering a credit card number or just before completing the rewards reservation) in a major U.S. city that has multiple H properties. Within that task, participants will select a hotel and room based on a combination of price and amenities. Each group will start at a different point:

Group 1	Start at H.com
Group 2	Start at non-branded search from Google (example: premium San Francisco hotel 4 star hotel).
Group 3	Start from a branded search from Google (example: H Hotel Atlanta)

Let the participants start where they would normally start: Because you'll select participants for different combinations of characteristics, expect that different types of participants are motivated to do different things. Briefly interview the participant at the beginning of the session to get some impression of how the particular participant approaches booking travel arrangements–especially accommodations–and let them perform the task within their own context. This way, in addition to getting a feeling for the overall usability of `www.H.com`, you can also identify usage patterns that could be further investigated in follow-on research. Finally, you will also get a better understanding of the traveler's thought processes and how H.com fits into that traveler's life.

Figure 5-5 Task description for an exploratory usability test

Having analyzed label usage, context, and location, you know that merely asking the participants to explain the label's meaning does not really suffice. Instead, you have to provide a task during which they are expected to use the label, and ascertain whether they notice, read, and use the label correctly (see Figure 5-5). In fact, the label is actually secondary to the task of putting images into collections that it supports.

The actual task, then, that exposes the label's usability is:

Arrange images into collections.

Notice that the task description does not even mention the label.

The label usage is explored indirectly by the test moderator while the task is performed. The test moderator must note where the participants look and so forth, and then question them during the debriefing session.

Having arrived at the correct task to meet your objective, let's classify it according to the four parts of a task as we outlined in the previous section. The table below shows a description for each of the four parts of a testing task.

TASK COMPONENT	DESCRIPTION
Task	Arrange images into collections.
State	Web site with six navigation tabs leading to different sections of the site.
Successful completion criteria	Participant finds the Organize link and then groups preloaded pictures into sets.
Benchmark	Participant puts more than two pictures into one set with no ''wrong turns.''

Ways to Prioritize Tasks

Now that you have reviewed an example of developing a task, the next issue is ascertaining what tasks you need to include. Due to time constraints, very rarely do you actually test the full range of tasks that comprise an entire interface, documentation, or both together. (It is impractical to conduct test sessions that last for days at a time, unless you are willing to commit an inordinate amount of resources.) Instead, you typically face a situation of testing a representative sample of the product's functions.

When choosing this sample of tasks, it is important that you exercise as many of the most important aspects of the product as possible and address all test objectives. Filter or reduce your task list to something manageable, while ensuring that you capture as many of the usability deficiencies as possible. The following list outlines some common methods you can use that prioritize or pare down the task list without needless sacrifice.

- **Prioritize by frequency.** Select those tasks that represent the most frequently performed tasks of your end user population. The most frequent tasks are the ones that the typical end user performs daily, possibly up to 75 to 80 percent of the time, when using the product. For example, if you were testing a word processing package, you would want to make sure that the end user could easily perform the following tasks before you concern yourself with the more esoteric tasks such as ''how to hide a comment that does not print out.''

  1. Open a file.
  2. Save a file.
  3. Edit a file
  4. Print a file.

Often, tests are filled with a series of obscure tasks that less than 5 percent of the end user population will ever find, never mind use. Why? Our theory is that the development team finds those ''five percenters'' the most interesting and challenging tasks to implement, because they are usually the leading edge of the product. Unfortunately, the typical end user does not share the developer's priority or enthusiasm for these obscure tasks.

If, after applying the ''75 percent usage guideline,'' there is still time to test more tasks, include tasks that at least 25 percent of your end user population perform regularly. Only when you are sure that the frequent tasks are covered should you include the less frequently performed tasks.

- **Prioritize by criticality.** Critical tasks are those that, if performed incorrectly or missed, have serious consequences either to the end user, to the product, or to the reputation of the company; for example, when the tasks result in a support line call, cause loss of data, or cause damage to the product or bodily harm to the user. In short, you want to make sure that you catch those tasks that result in the most pain and potentially bad publicity.
- **Prioritize by vulnerability.** Vulnerability in this case means those tasks that you suspect, even before testing, will be hard to perform or that have known design flaws. Often, the development team will have a good handle on this and, when asked, will voice concern for a new feature, process, interface style, section of a document, and so on. If so, include tasks in the test that address these major areas.

  Sometimes, developers pretend, in the name of ''being unbiased,'' that all functions work equally well (or poorly), and that none are particularly problematic. Whether for a well-intended or a less noble reason, they do not want known problems exposed during the test. Consequently, tasks that are obviously hard to perform and that represent whole components, web pages, or sections of a document are left out of the test and prove to be albatrosses much later when there is no time to fix them. To avoid that, use *your* critical judgment about which tasks/features are not quite worked out, are new or never-before-tested features, or have been difficult for in-house personnel to perform. If you are unsure, a human factors specialist can help determine the vulnerable aspects of the product by performing an evaluation. (An expert evaluation can also help you to tighten your test objectives in general.)
- **Prioritize by readiness.** If you are testing very late in the development cycle, you may simply have to go with functions that are ready

to be tested or forgo testing entirely. While this is not ideal, it is sometimes your only choice. You will not always have the luxury of waiting for every last component, screen, and user manual section to be completed. *Remember, it is always better to test something than nothing*.

Describe the Test Environment, Equipment, and Logistics

This section of the test plan describes the environment you will attempt to simulate during the test and the equipment that the participants will require. For example, you might want to simulate a sales office for a product that insurance agents use. Or, perhaps chemists use your product in an environmental laboratory. Or, suppose that you simply want to test the product in a very noisy, somewhat crowded office where phones are constantly ringing. Whatever the typical operating environment, try your best to simulate actual conditions. Not only does this help the participants to take on the role of actual end users, but it also means the test results will be a better predictor of the product's performance in the place where it is normally used.

The equipment described here only includes the equipment that participants will use. Examples of equipment are phones, computers, printers, and so forth. It is not necessary to describe data collection equipment or cameras you will be using to monitor the test. Figure 5-6 shows one example.

Explain What the Moderator Will Do

This section helps to clarify what you as a test moderator will be doing, and it is especially important when there will be observers of the test who are unfamiliar with the testing process. (See the example in Figure 5-7.) Specify when the test moderator will do something out of the ordinary that may lead to confusion. For example, sometimes it is unclear why and under what circumstances the test moderator is probing and intervening. This is especially

Test environment

We'll use a controlled setting to conduct the sessions. The study will take place at Acme Research in Fresno, California. There will be a testing room with a one-way mirror to an observation room.

Participants will use a Windows PC and Internet Explorer 6.0 with a high-speed connected to the Internet. The PC that the participant uses will also have Morae Recorder installed on it and a webcam attached. The webcam will capture the participant's face; the Morae software will record what's happening on the screen (and can collect other data). I will bring a digital voice recorder to the sessions to create a set of audio recordings for backup.

Figure 5-6 Test environment description of location and setup

Moderator role

I will sit in the room with the participant while conducting the session. I will introduce the session, conduct a short background interview, and then introduce tasks as appropriate. Because this study is somewhat exploratory, I may ask unscripted follow-up questions to clarity the participants' behavior and expectations. I will also take detailed notes and record the participants' behavior and comments.

Figure 5-7 Moderator role description

true when the test moderator may be role-playing or intentionally playing devil's advocate with an overly acquiescent participant.

List the Data You Will Collect

This section of the test plan provides an overview of the types of measures you will collect during the test, both performance and preference data. Performance data, representing measures of participant behavior, includes error rates, number of accesses of the help by task, time to perform a task, and so on. Preference data, representing measures of participant opinion or thought process, includes participant rankings, answers to questions, and so forth. The data collected should be based on your research questions. Sometimes these measures will have already been alluded to in a previous section of the test plan, such as the methodology section. You can use both performance and preference measures either quantitatively or qualitatively, depending on the test objectives. See Figure 5-8 for example measures for a test of a hotel reservations web site.

Listing the evaluation measures you will use enables any interested parties to scan the test plan to make sure that they will be getting the type of data they expect from the test.

The following is a sample of the types of measures you might collect during a typical test.

Sample Performance Measures

- Number and percentage of tasks completed correctly with and without prompts or assistance
- Number and type of prompts given
- Number and percentage of tasks completed incorrectly
- Count of all incorrect selections (errors)
- Count of errors of omission
- Count of incorrect menu choices
- Count of incorrect icons selected

Measures

To answer these questions:

- How easily and successfully do travelers get started with making a reservation on the site?
- Does the starting point make any difference in whether travelers are successful in reaching their goal on `www.H.com`? If so, what are the differences?
- What paths do travelers take to completing a booking?
- How well does the site support the paths and goals of the travelers? That is, how closely does the organization and flow of the site match travelers' expectations?
- What obstacles do travelers encounter on the way to completing a booking?
- What questions do travelers ask as they work through their reservation?
- How do travelers feel about how long it takes them to complete an online booking, both in the perceived amount of time and the number of steps?

I will collect both performance and preference data during the test sessions.

Performance:

- Errors of omission
- Errors of comission
- Number of tasks completed with and without assistance—I will track two levels of prompting when participants need assistance.

  None Participant completed a task without prompting.

  Try again Participant completed a task when asked, "Can you think of any other place to look?"

Preference:

- Appropriateness of site's functions to users' tasks
- Perceived amount of time and number of steps
- Ease of use overall
- Usefulness of terms and labeling

Figure 5-8 Sample measures for a test of a hotel reservation web site

- Count of calls to the help desk
- Count of user manual accesses
- Count of visits to the index
- Count of visits to the table of contents
- Count of ''negative comments or mannerisms''
- Time required to access information in the manual
- Time required to access information in online help
- Time needed to recover from error(s)
- Time spent reading a specific section of a manual
- Time spent talking to help desk
- Time to complete each task

Qualitative Data

- Think aloud verbal protocol
- Quotable quotes: (for example)
  - ''I loved it — when can I get one?''
  - ''You guys have done it again — you're *still* not listening to customers.''
  - ''Wow, I'm very, very impressed.''
  - ''Can I please leave now — keep my money and the product.''

Sample Preference Measures

Ratings and rationale concerning:

- Usefulness of the product
- How well product matched expectations
- Appropriateness of product functions to user's tasks
- Ease of use overall
- Ease of learning overall
- Ease of setup and installation
- Ease of accessibility
- Usefulness of the index, table of contents, help, graphics, and so on
- Help desk replies to inquiries
- Ease of reading text on the screen

Preference and rationale for:

- One prototype vs. another prototype
- This product vs. a competitor's product
- This product's conceptual model vs. the old model

Describe How the Results Will Be Reported

This section provides a summary of the main sections of your test report and the way in which you intend to communicate the results to the development team. For the report contents section, simply list the sections that will appear in your test report, as shown in the example in Figure 5-9.

For the presentation section, describe how you will communicate results to the development team both prior to and following the report. For example, you might hold an informal meeting with those on the critical path of the project just after the test is completed and prior to analyzing all the data. Then,

Report contents

I will deliver a draft of the final report to my point of contact at `www.H.com` that:

- Briefly summarizes the background of the study, including the goals, methodology, logistics, and participant characteristics
- Presents findings for the original questions to investigate
- Gives quantitative results and discusses specifics as appropriate to the question and the data
- Provides visuals of pages of `www.H.com` that are relevant to specific questions where they will help reviewers understand what we are talking about
- Discusses the implications of the results
- Provides recommendations
- Suggests follow-on research

H will review the draft and comment on it. I'll incorporate agreed changes and then present a summary of the findings in a meeting at H's headquarters

Figure 5-9 Description of what will be in the report

following completion of all analyses and the test report, you might follow that with a formal presentation to the entire project team, as well as other interested parties, management, and so forth.

Sample Test Plan

If we roll up the parts of the plan that we have included as samples above, the test plan deliverable comes to about 10 pages. To see the full plan, go to the web site that accompanies this book (`www.wiley.com/go/usabilitytesting`). The product being tested is a hotel room reservations system on a web site. There, you can view more examples of test plans, as well as download templates for test plans and other related deliverables in the file `Ch05 Hdotcom_test_plan.doc`.