1. Form a team of two or three people

2. Pick some program or system to study

It should be something that is useful but has a poor or tricky user interface. It should also be something for which you can find an appropriate test user, in particular, if an advanced application, then a knowledgeable user.

For example:

<table>
<thead>
<tr>
<th>Generic Unix Programs</th>
<th>Knome tools</th>
</tr>
</thead>
<tbody>
<tr>
<td>sort</td>
<td>project management (under Applications -&gt; Office menu)</td>
</tr>
<tr>
<td>du</td>
<td>disk usage analyzer (under Applications -&gt; System Tools)</td>
</tr>
<tr>
<td>grep</td>
<td>Kalarm (under Utilities -&gt; PIM)</td>
</tr>
<tr>
<td>find</td>
<td>Karm (under Utilities -&gt; PIM)</td>
</tr>
<tr>
<td>apropos</td>
<td>Other</td>
</tr>
<tr>
<td>man</td>
<td><a href="http://www.systransoft.com/">http://www.systransoft.com/</a></td>
</tr>
<tr>
<td></td>
<td>the breakfast menu at McDonald’s</td>
</tr>
<tr>
<td></td>
<td>Goldmine</td>
</tr>
</tbody>
</table>

3. Decide your methods

   a. what the tasks will be (see note below)
   b. what to look for (see note below)
   c. who will be the experimenter(s) and who will be the user(s)
   d. how much training or learning time you will give the user(s)
   e. how you will observe the users (see section 4.3.3. of the textbook for some options)
   f. what initial questions you will ask them (see note below)
   g. what the consent form will say (see attached)
   h. etc.

4. See the instructor to discuss your plans (preferably before the end of class)

5. Do the usability study

6. Write a report of your findings

Your write-up should be oriented towards a senior person who will make the decisions regarding the future of this software. Use the following format:

   Section 1: Background: Give a brief description of what the system is, including screen dumps and sketches as appropriate.
Section 2: Methodology: Explain what you did, noting in particular your choices of methods.

Section 3: Observations.

Section 4: Interpretation: Identify common and important problems and strengths of the system, generalizing from the specific observations.

Section 5: Design recommendations (do not do this now; this will be added in the next assignment).

Appendix: Raw data: All original observations or recordings should be included.

7. Prepare a presentation. Your team will have five minutes to present to the class on October 1. Probably overhead-camera slides are best, although the whiteboard and the projector are also available. If you chose the latter, e-mail the presentation to the instructor by 3:00 p.m. on the day of presentation.

Grading: 20 points (10 for doing all the work: 10, 8 for clearly explaining it, 2 for interesting findings)

Estimated time to complete: 3 hours
Due at the start of class, October 1
Background

How can we tell if a computer system is any good for a person to use? Most developers simply create the system, try it out themselves until they are satisfied with it, and then dump it onto the user audience. The result is usually a product that people have problems with.

One of the easiest methods for getting to "know the user" and for evaluating the human computer interface is through usability studies. Although these come in many flavors, they all require an observer to watch a typical user try the system out for a real task. It is surprising how many design flaws can be detected this way!

Usability studies are increasingly popular in industry. Many modern software companies now have usability labs staffed by HCI (Human-Computer Interaction) professionals whose job it is to find usability problems in products as they are being developed. Most labs contain all the equipment permanently in place (e.g., computers), and are instrumented with audio, video, screen-capture software, one-way mirrors, and so on.

Usability studies are extremely practical, and you can do them 'on the cheap' without these special usability labs. The simplest studies just require you: to pull up a chair next to a typical user; to watch them do their work (and perhaps having them explain what they are doing as they are doing it); to jot down any noteworthy events that occurred; and to listen to the user's comments.

Scenario

You work for Usability Inc., a consulting firm that specializes in evaluating interfaces. You and your team have been contracted to do a usability study of a system. Your deliverable will be a report written for the Vice President in charge of that system's use and redevelopment. Your report will describe how you went about looking for design problems, what problems you saw, and what changes you recommend. Depending upon how convincing you are, the VP will authorize changes in the upcoming version.

Note on task selection

Usability studies require an observer to watch someone go through the paces with 'typical' tasks. It is your job as experimenter to prepare a set of example tasks ahead of time that the subjects will try to perform. These tasks should be realistic ones that typical users would try to do with the system. But how do you discover what those typical tasks are?

The first way is to let subjects use their own real tasks. To do this, you would have to solicit subjects who have a real need, and ask them if you could watch them do their tasks.

The second way is to ask a random sample of people who are using the system what they typically do with it, and then generalize those as tasks to give to subjects.

The third way is for you to use the system, and contrive a few sample tasks through intuition. Although this will not produce a set of reliable tasks, you may not have any other choice.

Note on deciding what to look for

It is important to decide what to observe. The easiest thing to watch is progress on the task. You may also want to focus on wasted time, or on errors, or on confusions, or on mental model formation. This
last is especially important for users trying novel interfaces, where we want to see how well the user is able to form an understanding of the interface.

You can gain a sense of a person's initial conceptual model of the system by having them explain each screen as it appears, including what interface components they see and what they think they can do with them. You are looking for places where the model is incorrect or undeveloped. For example, people may not understand the meaning of labels and icons, what they are supposed to do, and how they are supposed to use them. Some of these problems are related to the lack of meaningful visual affordances, constraints, mapping, and so on.

Usually all your subjects should begin with this step. Using this information as a baseline, you can see how a person's conceptual model develops (correctly and incorrectly) during system use merely by asking them to re-explain the display after major dialog steps (e.g., after reading documentation or after doing a transaction) or at the end of their session. Note that this means you are actively intervening in a person's session, for you are disrupting them in the middle of their task, and their act of explaining the screen to you may result in extra learning by them. Thus you should do this with restraint, and only at opportune moments.

**Note on understanding the user’s background and beliefs**

A user's interactions with interface can be greatly affected by their background and attitude, and you will need to find out at least something about these before you start. In particular, you should learn about their prior experience (with computers, with the operating system, with the system being tested, and with the tasks), and about their expectations (about whether they expect the interface to be a good or a bad one, and whether they expect it to be usable immediately or to require some exploration or training first).

Often it is appropriate to create a written questionnaire so that you can be sure to get all this information from each user; but for this assignment it is fine to just ask.

(This assignment is based on one by Saul Greenberg of the University of Calgary.)