1) Three non-desktop interfaces

**Steering column of a car**

The steering column is used by anyone who drives a car or truck. The primary functions are the same in any vehicle, making its use very similar in virtually every context.

Strengths:

1) When a user sits in the driver’s seat of a vehicle, their first instinct is to grab the steering wheel. Its shape affords grabbing, so there is no confusion over how to operate it.

2) The steering wheel’s location directly in front of the driver is a clue to its importance in operating the vehicle. Even if you knew nothing about how to operate the vehicle, you could probably guess that you need to know how to use the wheel in order to drive it. If the wheel had been something smaller and positioned between the seats, this may not be so obvious.

3) Placing controls for things like headlights, turn signals, and windshield wipers on the sides of the steering column reduces the effort required by the driver to find these controls. These systems are used frequently enough that they need to be in a position that allows the driver to operate them without having to take his or her eyes off the road.

Weaknesses

1) The addition of other controls on or around the steering wheel is both a good thing and a bad thing. Some cars today have controls for the radio on the steering wheel. While it is good to prevent the driver from having to reach for them, they could easily be confused with other controls. For example, some cars have up and down arrow-shaped buttons for that operate the cruise control, while others have similar-shaped buttons that control the volume of the radio. It would be easy to accidentally set the cruise control when you intended to decrease the volume of the radio, potentially leading to an accident. Rather than eliminating the extra controls, the automakers should agree on some standards. One possibility is to have the buttons for the cruise control on the left side of the wheel and the buttons for the radio on the other side of the wheel.

2) All cars and trucks have windshield wipers, but the controls for them vary by automaker. On some cars, you have to twist a lever to turn them on, while on others you have to push a lever down. Some cars use a twisting motion on the lever to set the speed, while others have you pull down to different notches to set the speed. Once again, the best way to fix this problem is consistency. A set of standards would make it so drivers
don’t even have to think about how to operate the controls; they could just rely on their experience with previous vehicles.

3) Almost all steering columns allow the driver to adjust the wheel to a comfortable position. Unfortunately the mechanism for making this adjustment is often difficult to find. I have seen cars that had a button on either side of the wheel, but it is rarely labeled. If it is labeled, you cannot easily see the label because of its position behind the wheel. This problem could be solved with a label on the wheel indicating where the adjustor is. Placing the button at the base of the steering column was a good design decision because it keeps the driver from hitting it accidentally, but the instructions for operating it should be clearly visible.

Remote controls

Remote controls are used in social settings to control a TV, and in a multitude of work settings (for example, to control a projector or advance slides in a presentation).

Strengths

1) Most remotes for a particular type of device have a number of buttons that you can always count on being there. For instance, a TV remote will have a power button, channel up/down, volume up/down, and a number pad. This allows the most basic functions of the TV (the same is true for remotes for other devices) to be performed with the remote without ever having to read a manual.

2) The shape of most remotes is generally the same. Except for a few oddities, most remotes are rectangular. This makes remote controls easy to identify (for example, if you are trying to find the remote on a cluttered coffee table), and also affords pointing the device at whatever you are trying to control. Remotes can usually only be held in two ways that make sense. One of these has the infrared diode pointing toward your body, while the other has the diode pointing away from your body. One can use this information along with the direction the button labels are facing to easily determine the correct way to hold the remote.

3) The power button, which is probably the most important button on the remote, is usually clearly distinguished from all other buttons. Whether it’s a different color, larger size, or separated spatially from the other buttons, your eye is usually drawn directly to the power button. There are some extremely non-tech savvy people out there who might press that button first for the simple fact that it looks different from all the others.
Weaknesses

1) There are no standards about where common buttons are located on a remote. I've seen many TV and radio remotes, each of which has a set of channel (or radio station) up/down buttons and a set of volume buttons. Often time these sets of buttons look the same, and they are usually right next to each other, but you never know which set is which without reading the labels every time. Because of this, you can constantly lower the volume when you were trying to go to the previous channel, or vice versa. This would not be a difficult thing to remember, except that you could have a number of remotes in your living room that have different configurations. A simple way to solve this problem: use different-shaped buttons. Use +/- buttons for volume and up/down buttons for channels, or vice versa. If you can feel the shape of the button, you should be able to accomplish the simple task of adjusting the volume or changing the channel without any extra effort.

2) There are too many remotes that have one rubber button that controls two functions. Depending on which part of the rubber you hit, you could perform one function or another. This is very common with volume up/down, channel up/down, and to a lesser extent the various inputs on a TV. When I see one button-shaped object on my remote, I assume that it has one function. You should not have to apply pressure on two different parts of the same button to perform two independent operations. Making the channel up button completely separate from the channel down button would remove any ambiguity.

3) Some remotes have a cover that hides some of the lesser-used buttons from the user. This seems to be a poor way to place extra emphasis on the more commonly used buttons. There are no clear indicators that any buttons are hidden, and only a small notch on the side of the remote that indicates that there is a cover to be removed. A better design might have simply made these hidden buttons smaller, rather than cover them up. Or, if the designers were adamant that they be covered up, a notice on the remote indicating that there were more advanced controls under the cover, along with instructions for how to remove the cover would have been useful.

iPod (classic version)

The iPod is used entirely in a social context, although each device is typically used by one person. The designers apparently relied on this fact when they made the decision to allow users to hide some menu items from the main menu of the iPod, and provided no clear way to unhide them. They assumed that anyone who knew how to hide a menu item would also know how to unhide them if necessary.
Strengths

1) The best feature of the older iPod models is the scroll wheel. It takes advantage of the clock analogy to allow users to navigate through the menus of the device without any explicit instructions. The analogy is that moving forward in time (ie, moving clockwise) is the same as moving forward on a menu. Since you start at the beginning of the list, it is clear that forward on the menu means downward on the list.

2) Menus are also made easier to navigate by the one window drill down design pattern. Rather than showing additional information on the screen when you select an item, you are taken to an entirely new screen, with the name of the parent menu at the top of the screen so the user knows which screen they were looking at previously. On a screen as small as the iPod’s, this design pattern keeps the content on the screen uncluttered and easy to read.

3) It can be difficult to keep track of all your music if you have thousands of songs on your iPod. This is a problem for what is primarily a music-playing device, as users expect to be able to find any of their music and play it with little difficulty. This is why the ability to view songs by artist, album, and genre is such a useful feature. The iPod does not allow you to enter a song or artist name to search for because it lacks a keyboard, so the designers added these multiple ways to list songs to make manual searching easier on the user. While it is not quite as useful as typing in a search term and running an automatic search, it is a good alternative that makes the most out of the simple control system (scroll wheel and center button).

Weaknesses

1) While the scroll wheel is a nice feature, it can be difficult to scroll through a very long list. The iPod has a quick-scroll feature that lets you scroll through letters of the alphabet rather than scrolling one item at a time. The problem with this feature that there is no way to manually begin quick scrolling; it kicks in automatically after you’ve been scrolling quickly for some period of time, and stops some period of time after you’ve stopped scrolling. This makes it too easy to accidentally scroll past what you are looking for in a long list. This could be fixed by allowing the user to activate and deactivate quick scrolling manually, for example by holding the center button for a few seconds before and after scrolling with the wheel.

2) Another minor problem with the scroll wheel is that it can bee too sensitive at times. Occasionally users accidentally scroll up or down one item when they lift their finger off the wheel. When this happens, it is usually because he or she just finished scrolling for an extended period of time down a long list, and then lifted their finger suddenly upon reaching the target. The simplest way to fix this problem would be to decrease the sensitivity of the
wheel. However, this could make the scrolling feel sluggish to users who expect it to move very fast. A better solution would be to make the sensitivity adjustable by the user, who could decide whether or not they wanted to increase accuracy (i.e., decrease sensitivity) at the cost of some speed.

3) Even though the iPod does a lot to make songs easy to search through, it does a surprisingly poor job of handling playlists. It allows you to add songs to and play songs from an “on the go” playlist, but you cannot create multiple playlists with the iPod; you must connect it to Apple’s iTunes to do so. While it makes sense that you cannot name a playlist due to the lack of a keyboard, you should at least be able to create multiple playlists and make a distinction between them without hooking the device up to a computer. One way this could be done is by allowing the user to create new playlists named as “playlist 1”, “playlist 2”, etc. Then instead of adding a song to the “on the go” playlist, the iPod can ask the user which playlist to add it to. You would still have to connect the iPod to a computer to rename the lists, but at least you have your songs organized the way you want.
2) Learning to listen to users

I interviewed user X, user Y, and user Z about their experiences using computers. User X is a novice user who uses computers mostly to check email and do basic Internet browsing. User Y is a pretty average user; he uses email, Internet browsers, banking programs, word processors, and another of other applications. User Z is a more advanced user. He uses his computer for all of his work, and generally knows how to get things done with them. If he doesn't know how to do something, he usually knows where to look to figure it out.

**User X**

User X loses her ability to use a website whose buttons or links are rearranged. This occurs because she associates the actions she wants to perform with locations to click on the screen. If that location looks different from one session to the next, she gets frustrated and asks for help. Her opinion is that the pages she uses should never be changed. While this is probably unrealistic, the designers of the pages should keep this type of user in mind when they change things. If they move things around, they should make it obvious that instead of clicking a button in the top right corner to perform operation X, you need to click a link in the bottom right corner to perform operation X.

Another problem encountered is Windows error messages. The machine used is fairly old and not very well maintained, and thus is sometimes prone to odd errors. User X has encountered numerous error messages that “have a bunch of numbers and a bunch of words I don’t understand.” Not only did the error message not describe the problem in words she could understand, but it also did not tell her what she could do to fix the problem. After she reads the error message, she normally clicks an OK button and never thinks twice about it. She says that the solution is not to show these messages at all. I think a better solution would be to provide better information. Some of that jargon is necessary for tech support to diagnose the problem, but the user shouldn’t be left in the dark. At the very least, the message should provide a link to a resource that can give more information, and it should tell the user what to do, whether it is “Please restart your computer” or “Please call Microsoft tech support.” If the message does not tell the user what to do about the error, it is likely that the error will keep occurring.

The problem that annoyed user X the most was sending email attachments. Whenever she receives a funny picture or a video as an attachment, she saves it to her computer desktop and then opens it. If she likes it, she sends it to a number of her friends. Sometimes instead of forwarding the original email, she decides to create a new message and send the files that way. However, she always has trouble attaching her files to the message. When she presses the “attachment” button, she sees a new window asking her to select the files she wants to send, but she doesn’t always see the files she wants to send. The problem is that she is looking at the wrong directory. Her browser is configured to save everything on her desktop, but sometimes the window that asks her to attach the files is showing a directory other
than her desktop and she does not know how to navigate to where she needs to be. She says that she would like to be able to click and drag the files directly into the email. This sounds like a very reasonable solution, especially considering you can do the same to move files into other folders on your computer, or into a word processor.

**User Y**

User Y knows how to navigate around the Internet, but occasionally has trouble finding the content he is looking for on a website. As an example, he recently missed a show that he wanted to see on the National Geographic channel. He went to the National Geographic website to look for information. The show was about some species of fish, so he clicked on a sidebar link that said “Animals”. From there he saw lots of information about other types of fish, but nothing about the one he was looking for. When he walked me through this process, I noticed that he was looking at information on the National Geographic Magazine, not the National Geographic Channel. There was a link labeled National Geographic Channel a little lower on the page. When I pointed this out, he asked me why it wasn’t at the top of the page? He makes a very good point. The National Geographic Magazine is similar to, but not the same as, the National Geographic Channel. He thinks that there should be a link at the top of the page to direct him directly to the content for the channel. He described it as being similar to the link to a disambiguation page at the top of a Wikipedia article.

One major source of frustration to user Y was the transition from Windows XP to Windows Vista. He described a number of problems, but one of the ones I found most interesting was when he described the process of trying to remove some of the programs that come preinstalled on a Hewlett-Packard computer. He was looking for “Add/Remove Programs” in the Control Panel and could not find it. He read through the entire list several times and could not find it. Eventually he discovered that what he needed was actually in “Programs and Features” in the Control Panel. I asked him why he didn’t think to check that earlier since it at least had the word “Programs” in it. He said that he was expecting to see the word “remove”. When asked how he thought this type of issue could be avoided, he said “if it works, don’t fix it.” His opinion is that they shouldn’t have changed the name of the item in the Control Panel. People have gotten used to looking for “Add/Remove Programs” when they want to uninstall a program, and now they have to train themselves to look for something else. Furthermore, “Add/Remove Programs” was at the top of the list, whereas “Programs and Features” is toward the bottom of the list. This does nothing but add to the confusion.

Another problem he had was also related to consistency between versions of an application. He complained about the new interface of Microsoft Word. His biggest problem was the replacement of the traditional menu bar (that included File, Edit, etc) by the tabbed menu system that appears at the top of the window. He got used to clicking on File and then quickly dragging the mouse down a certain distance to save a file. He felt like he had to learn the brand new interface just to
perform the same basic operations that he used to. He also said that it didn't make sense to remove that menu bar since almost every other program he's ever used had the same interface to basic operations like save, open, and exit. I think the solution to these types of problems is a more gradual change in interfaces. The designers should make a few small changes to the UI and release that version of the application. Then for the next version, implement a few more of the changes. This allows the users to get used to the changes more gradually. They can learn a few changes at a time, rather than falling into a situation where everything is brand new and they are unable to do anything that they used to be able to do.

User Z

User Z recently started using Linux, but has complained about the difficulty of installing applications. He says that some things can be installed automatically from a graphical interface, but not everything he needs is available from that system. Other applications can be downloaded and installed automatically with a command from the terminal. Still others must be downloaded and compiled manually. This last set is almost impossible for him to do because he has very little experience compiling programs. While many of them come with installation instructions, he says they usually just list the commands that need to be executed without any explanation of what to do if any of the steps fails (a very common problem, he says). When asked about alternatives, he said that it would be nice if all these applications could use a system similar to the Windows installers. He likes how these walk you through the entire installation process. I agree that this would make more sense, since the end user normally wants to be able to get the application running with a few clicks, without having to manually resolve any dependency issues.

While learning how to use the Linux terminal, user Z has had to make frequent use of manual pages. He describes what he calls a “stupid problem” in that he keeps forgetting how to get back to the command prompt when he is finished viewing a manual page. He often tries ESC and ‘e’ (for “exit”) before hitting ‘q’ for “quit”. By itself, this would be just a minor annoyance. However, in the context of trying to figure out how things work while struggling to remember which key to press to exit the manual page, it was very frustrating. I think a way to resolve these issues would be to show a help screen when the user types a ‘?’ character. Currently the question mark is used for a search, but I think it makes more sense for it to be a gateway to a list of commands that can be run in the man program. This help screen would tell users how to do the basics like navigating on the page, closing the program, and searching. It could also contain more advanced commands that are not used as often, but that power users might need to know.

User Z is a former Mac user. One of the things he didn’t like about the operating system was the dock. When he browses the Internet, he routinely has several browsers open at the same time. He doesn’t like that the dock gives no indication of how many of them are open. Whether he has one instance open or 10 instances open, the dock has a small blue dot below the icon for the browser. While
there is a Window button in the mile high menu that lets you choose the window, a good solution to this problem would be to show all instances of the currently active application in the dock. This would provide a good balance between those who are used to the traditional Mac dock and those who are migrating from Windows or Linux who are used to seeing each instance open at the bottom of the screen.
3) Power Law of Practice

The person I had run through these trials is an experienced typist. It took him some time to correctly type in the correct letters on the first trial. He said he was taking it slow the first few times until he could. He made a few failed attempts between recorded trials two and three because he said he was trying to type too fast. After that, he was pretty consistent. He said his strategy was to break the sequence down into two to three character chunks. He typed each chunk very quickly and had a brief pause between chunks. In the middle range of trials, he spent more time between chunks thinking about the next pattern. In the final trials, he was able to minimize the thinking time between entering these chunks.

The equation for the trendline shows that the learning constant is approximately .46.

Source code for this experiment is below.
import java.io.*;

public class Practice {
    private static final int TRIALS = 20;
    private static final String TARGET = "stuvwxyz";
    private static final String REVERSED_TARGET = "zyxwvuts";
    private static long[] times;
    private static long start;
    private static long end;

    public static void main(String[] args) {
        times = new long[TRIALS];
        try {
            for (int trial = 0; trial < TRIALS; trial++) {
                String line;
                InputStreamReader reader = new InputStreamReader(System.in);
                BufferedReader bufReader = new BufferedReader(reader);
                startTrialInXSeconds(trial, 5);
                start = System.currentTimeMillis();
                line = bufReader.readLine();
                end = System.currentTimeMillis();
                if (line.equalsIgnoreCase(REVERSED_TARGET)) {
                    times[trial] = (long)(end - start);
                    System.out.println("Good. "+ (TRIALS - trial - 1) + " trials left.");
                } else {
                    System.out.println("That is not correct please try again(" + line + ")");
                    trial--;
                }
            }
            System.out.println("RESULTS: ");
            for (int trial = 0; trial < TRIALS; trial++) {
                System.out.println("Trial " + trial + ": "+ times[trial]);
            }
        }
    }
}
```java
private static void startTrialInXSeconds(int trial, int X) throws InterruptedException {
    System.out.println("----------------------------");
    System.out.println("Trial "+(trial+1)+" / "+TRIALS+":
Type the last 8 letters of the alphabet backwards");
    System.out.println("trial starts in "+X+"...");
    for (int i = X-1; i > 0; i--)
    {
        Thread.sleep(1000);
        System.out.println(" "+i+"...");
    }
    Thread.sleep(1000);
    System.out.println("Go!");
    System.out.println("----------------------------");
}
```
4) Choice Reaction Time

a) If we assume error free behavior and that the user knows the target item, the time to select an item from the menu is dominated by the movement time: 
\[ MT = a + b \log(2A / W) \]

If we assume that all menu items are the same width, the target’s width is

\[ W = M/12, \text{ where } M \text{ is the width of the menu.} \]

If we also assume that the item we choose will be near the middle of the list, the distance the mouse needs to travel can be expressed as 
\[ A = d + M/2, \text{ where } d \text{ is the distance between the current mouse position and the edge of the menu.} \]

This gives:

\[ MT = a + b \log(2 * (d + M/2) / (M/12)), \text{ which can be simplified to:} \]

\[ MT = a + b \log(2d / M+1) \]

b) The average distance required to move the mouse to one of the first four dynamic menu items is 
\[ Ad = d + M/6 \text{ (halfway through the dynamic area),} \]

and the average distance required to move the mouse to one of the static menu items is 
\[ As = d + 5M/8 \text{ (halfway through the static area).} \]

The amount of time required to select an item given a probability \( p \) of selecting an item from the dynamic area is therefore:

\[ MT = p(a+b \log(24*Ad / M)) + (1-p)(a+b \log(24*As / M)), \text{ which can be simplified to:} \]

\[ MT = a+b(p \log(Ad / As) + \log(24*As / M)), \]

where \( Ad = d+M/6 \) and \( As = d+5M/8 \)

i) When \( p = 0.5 \):
\[ MT = a+b(0.5 \log(Ad / As) + \log(24*As / M)) \]

When \( p = 0.75 \):
\[ MT = a+b(0.75 \log(Ad / As) + \log(24*As / M)) \]

When \( p = 0.9 \):
\[ MT = a+b(0.9 \log(Ad / As) + \log(24*As / M)) \]

Since \( Ad / As < 1, \log(Ad / As) < 0. \) This implies that \( MT \) decreases for larger values of \( p \).

ii) The minimum choice time would be the scenario where the probability of selecting an item from the dynamic area is 100%:
\[ MT = a+b(\log(Ad / As) + \log(24*As / M)) \]

iii) Since the choice time increases with more selections from the static area of the menu, a split of 0 / 100 would result in the maximum choice time.
c) Hick’s Law does not perform well in scenarios where the person must make a decision among items that are considered independently. That is, if eliminating one option does not eliminate any other options, an analysis using Hick’s Law will not provide accurate results. An example of this is when someone must choose one specific item from a randomly ordered list. The reason this is the case is because Hick’s Law assumes that people eliminate options logarithmically. However, this is not possible to do when each item must be considered individually, and the elimination of one provides no new information about the other items.